CITY UNIVERSITY LONDON

Visual Analytic Extraction of Meaning from Photo-Sharing Services for Leisure Pedestrian Routing

Alexander Kachkaev

alexander.kachkaev@city.ac.uk

Thesis submitted in fulfilment for the award of the degree of

Doctor of Philosophy in Geographical Information Science

Supervisors: Professor Jo Wood, Professor Dinos Arcoumanis

giCentre, School of Mathematics, Computer Science & Engineering, City University London

December 2014

Submitted December 2014

Viva voce May 2015

Minor edits for grammar and style August 2015

Abstract

Present-day routing services are able to provide travel directions for users of all modes of transport. Most of them are focusing on functional journeys (i.e. journeys linking given origin and destination with minimum cost) and pay less attention to recreational trips, in particular leisure walks in an urban context. These walks have predefined time or distance and as their purpose is the process of walking itself, the attractiveness of chosen paths starts playing an important role in route selection. Conventional map data that are informing routing algorithms cannot be used for extracting street attractiveness as they do not contain a subjective component, or in other words, do not tell whether or not people enjoy their presence at a particular place. Recent research demonstrates that the crowd-sourced data available from the photo-sharing websites have a potential for being a good source of this measure, thus becoming a base for a routing system that suggests attractive leisure walks.

This PhD research looks at existing projects, which aim to utilize user-generated photographic data for journey planning, and suggests new techniques that make the estimation of street attractiveness based on this source more reliable. First, we determine the artefacts in photographic datasets that may negatively impact the resulting attractiveness scores. Based on the findings, we suggest filtering methods that improve the compliance of the spatial distributions of photographs with the chosen purpose. Second, we discuss several approaches of assigning attractiveness scores to street segments and make conclusions about their differences. Finally, we experiment with the routing itself and develop a prototype system that suggests leisure walks through attractive streets in an urban area. The experiments we perform cover Central London and involve four photographic sources: Flickr, Geograph, Panoramio and Picasa.

A Visual analytic (VA) approach is used throughout the work to glean new insights. Being able to combine computation and the analytical capabilities of the human brain, this research method has proven to work well with complex data structures in a variety of tasks. The thesis contributes to VA as an example of what can be achieved by means of the visual exploration of data.

Acknowledgements

First of all, I would like to express gratitude to my family for making me who I am and especially for persuading me to continue education outside of the home town. It is impossible to imagine how much different my life would be if I was not pushed out of a 'comfort zone' a few years ago did not take an opportunity to study in Saint Petersburg and in London.

My PhD research would not be possible without a scholarship from the World Cities World Class University Network (`http://www.wc2network.org/`). I am thankful to its organisers and to the members of the recruitment committee, who chose me among other candidates for the funding. Without this decision it would be hard for me to make an attempt of contributing to Information Science.

I would especially like to thank my first supervisor, Professor Jo Wood for literally making my PhD possible. Without his initiative to start curating my project half a year after the beginning of my research programme, I would not be able to obtain so many new skills and probably would never finish writing the thesis. Jo's wise and broad view on many things positively influenced my reasoning, creative thinking and a sense of aesthetics. His practice of having frequent and productive meetings with research students has shaped all my work.

I am also indebted to my second supervisor, Professor Dinos Arcoumanis, for his belief in my research, for finding time to listen and discuss my ideas and also for helping me with sharing the outcomes of the research with members of the WC2 network.

Special thanks go to my colleagues at giCentre (`http://www.gicentre.net/`), from whom I learned a lot both explicitly and implicitly. In particular, I owe a huge debt of gratitude to Professor Jason Dykes for co-authoring one of my papers and also for inspiring everyone around with his incredible passion for work.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

In recent years the government and local authorities have taken a number of initiatives that aim to encourage walking in the UK. They mainly include making street infrastructure more suitable for pedestrians (Department for Transport 2011; MVA Consultency 2010), improving navigation by providing more information using maps and signs (Transport for London 2012; Woodhouse 2012) and also promoting walking as a physical activity and a part of a healthy lifestyle (Walk4Life 2012; WalkEngland 2012; WalkLondon 2012; Ramblers' Association 2012). A number of routing services (e.g. *Google Maps*[2], *Mapquest*[3], *TfL journey planner*[4], *Walkit*[5], etc.) help pedestrians get turn-by-turn directions for their journeys before making trips, thus also encouraging people to walk more. Most of the services are designed for finding directions between the given points with minimum cost (time or distance), thereby support walking for transportation purpose (*functional* walking). Meanwhile, route planning for *recreational* (or *leisure*) walking is not presented in these services with minor exceptions despite being rather popular, especially among the tourists (Ramblers' Association 2010). Unlike functional walking, recreational walking implies a more complex combination of factors

[2] `http://maps.google.com/`
[3] `http://mapquest.com/`
[4] `http://journeyplanner.tfl.gov.uk/`
[5] `http://walkit.com/`

that inform the selection of a particular route, many of which have a psychological nature and relate to human perception of space (Davies, Lumsdon and Weston 2012). One of the most hard-to-formalize factors that a person can be considering when planning a walk is the attractiveness of areas that appear on his or her way. A reliable approach to formalising this factor and its embedding into a pedestrian routing system could be found useful by millions of people planning leisure walks all over the world.

There is a significant difference between the planning of recreational walks in rural in urban areas. While rural footways are often designed for recreational walking, an urban street network in general has a functional purpose, thus potentially making the automation of choosing attractive paths in cities a more complex task. Previous research of data available from various sources of user-generated content shows that photo-sharing services can possibly form a reliable measure of popularity and attractiveness of space. Data from some photographic services has already been successfully used for detection of landmarks or advising tourists to visit the most popular places in a particular area (Andrienko et al. 2012, 2009; Baeza-Yates 2009; Dykes et al. 2008; Kisilevich et al. 2010; Purves, Edwardes and Wood 2011). These studies have demonstrated the existence of patterns in spatial and temporal distributions of photographs shared by internet users on Flickr and Panoramio and have proved the ability of such datasets to locate popular and attractive places in cities.

The idea of using the density of geotagged photographs as a measure of attractiveness of urban streets is based on the peculiarity of the process of photography sharing. In order for an image to appear on a photo-sharing website it must be taken and then uploaded by a user. Both of these actions are voluntary and due to the human psychology tend to happen when a person finds something interesting that is worth showing to others. When such behaviour is repeated in a large group of people, emerging patterns in distributions of photographs can be potentially turned into a measure of attractiveness of different places and streets. This feature of collectively gathered geotagged photographs suggests a study that looks into ways of their use in a routing algorithm for leisure walks.

1.2 Problem definition

Consider a network of walkable paths represented by a static graph

$$G = (V, E)$$

where V is a set of vertices (road intersections) and E is a set of edges of the graph (road segments). Every path can be presented as

$$P = v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \ldots \xrightarrow{e_{k-1}} v_k$$

or

$$P = e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_{k-1}$$

and is characterised by its weight

$$\omega_u(P) = \sum_{i=1}^{k-1} \omega_{u,e_i}$$

where ω_{u,e_i} is the cost of moving between nodes v_i and v_{i+1} via e_i for traveller (or traveller type) u. The value of the cost is non-negative. Distance and travel time are the simplest examples of such weights. In some applications the graph can be directed, so that $\omega_u(v_i, v_{i+1}) \neq \omega_u(v_{i+1}, v_i)$, but for the pedestrian routing the weights are usually equal in both direction. This is due to the nature of the subject: walking is not affected by traffic congestion or other factors that can change the weight of an edge based on the direction. Exceptions to this rule may occur, but such cases are not included into the focus of this research (e.g. when the slopes of the paths are considered, walking uphill may be with higher ω than walking downhill).

In a common scenario of solving a *single-source shortest path problem* (SP problem), e.g. with Dijkstra's algorithm, Bellman–Ford algorithm, Floyd's algorithm, etc. (McHugh 1990), the goal of a system is to find the shortest route (i.e. a path that has the smallest overall weight) $P'(v_{start}, v_{end})$ such as

$$\omega_u(P'(v_{start}, v_{end})) = \sigma_u(v_{start}, v_{end}) = min\{\omega_u(P) : P \text{ from } v_{start} \text{ to } v_{end}\}$$

$\sigma_u(v_{start}, v_{end})$ is the minimum theoretical cost of moving from v_{start} to v_{end} for traveller u.

If the weight of an edge is a combination of several factors, the task of path finding can be still transformed into a standard SP problem. This approach applies to functional walks.

A leisure walk through the attractive places is characterised not only by its cost, but also by some accumulating value of *gain*, which depends on the subjective measure of attractiveness for the chosen streets. Imagine a continuous geographical distribution $\mathbb{D}_u(x, y, \mathbb{a}_u)$, where $\mathbb{a}_u$ is a hypothetical ground truth for the value of attractiveness that a person u would assign to each point in space (x, y). Then, the value of attractiveness this individual would assign to a street segment $e \in E$ would be

$$\mathbb{A}_{e,u} = \mathbb{M}_{cont}(\mathbb{D}_u)$$

where $\mathbb{M}_{cont}$ is a function that maps a continuous distribution of attractiveness scores to scores assigned to the network edges.

The overall gain of an arbitrary walking route is

$$\mathbb{\Gamma}_u(P) = \mathbb{G}_u(\mathbb{A}_{e,u} : e \in P\})$$

where $\mathbb{G}_u$ is the accumulating function. This function is not necessary a sum of gains $\mathbb{A}_{e,u}$ for segments $\{e\}$, which a person u has chosen, because, for example, walking through some totally unattractive places and then visiting a very attractive street may be subjectively considered not equally beneficial as choosing a less diverse variety of streets, even if the total attractiveness score is equal in both cases.

Another difference between the functional (shortest path) walks and the leisure walks is that the latter ones can also be characterised by some predefined available budget ω'_u. This can be the amount of time a person expects to spend on a stroll, or the distance that he or she aims to travel. Thus, an optimal attractive leisure walk P' with the given ω'_u, v_{start} and v_{end} can be described as follows:

$$\begin{cases} \omega_u(P') - \omega'_u \geq 0 \\ P' = \underset{P(v_{start}, v_{end})}{\arg\max} \{\mathbb{\Gamma}_u(P)\} \end{cases} \tag{1.1}$$

Because the value of gain $\mathbb{\Gamma}(P')$ depends on the person's subjective distribution of street attractiveness, which cannot be extracted or predicted with any known technology, the problem of finding the *most optimal attractive leisure walk* does not have a solution. However, it can be attempted to substitute $\mathbb{\Gamma}_u(P)$ with some similar measure $\Gamma(P)$, if there is a reasonable correlation between the values of gain for the same chosen routes.

$\Gamma(P)$ can be extracted from the opinions of a group of people, assuming that the personal view on 'street attractiveness' does not significantly differ between individuals. Such approach has been used by mySociety (2009) to measure the attractiveness of different places in the UK. The method can be characterised with a high accuracy as people are surveyed about the attractiveness directly. However, it requires an active input from users and therefore may become impractical for covering large areas with data having a street-level granularity.

Previous research (Hochmair 2010) demonstrates that there is a correlation between the spatial density of photographs in crowd-sourced collections of images and the locations of scenic (attractive) routes, which suggests that these data can be used as an approximation for $\mathbb{\Gamma}_u(P)$.

It can be assumed that a geotagged photograph is a *positive vote* for a specific place to be attractive, as it is perceived by a particular photographer. Therefore, the distribution of a person's individual view on street attractiveness $\mathbb{D}_u(x, y, \mathbb{o}_u)$ can be represented as $\Phi_u(x, y, \phi_u)$, which is a distribution of positive votes for area attractiveness (ϕ_u is the density of photographs at a particular place by a particular user). When multiple Φ_u are combined into some distribution Φ, it can be attempted to estimate the approximate value of street attractiveness as measured by the given population of users:

$$A_e = M_{cont}(e, \Phi)$$

There may be a variety of ways to combine individual Φ_u into Φ. For example, the contributions by the professional photographers or knowledgeable people such as tourists guides can

be given higher credibility and consequently stronger influence on A_e. The difference in the number of images taken by a particular user at a particular location may also matter: the higher number of 'votes' a person has left at a street, the more attractive he or she potentially finds it.

Inclusion or exclusion of various factors into consideration establish the photographers' activity model and thus determine the nature of M_{cont}. Complex models have a potential to generate more accurate estimations of attractiveness scores, but at the same time are more difficult to establish correctly. For example, if a model suggests that the photographs by a professional tourist guide should influence street attractiveness scores more than those by a casual photographer, the model must provide means to distinguish different types of users with minimum number of errors. Failing to reliably do so may result far less accurate attractiveness scores than the ones obtained with a simpler activity model.

The content of the photographs is also a part of a model. If no constraints are provided here, the estimated values of street attractiveness may be irrelevant simply because large numbers of unsuitable geotagged images are counted as 'votes'.

For this work, it was decided to define the photographers' activity model using the following rules and assumptions:

> Distribution Φ only consists of photographs that are taken by the pedestrians.
>
> People only share photographs of attractive places, which constantly present at their locations (temporal structures are not captured).
>
> A single person can leave only one 'vote' at each location.
>
> A photographer's reputation is not taken into account.

These rules are plausible enough to justify the exploration of the real photographic datasets. The fact that the model simplifies some aspects of the real photographers' behaviour reduces the probability of false assumptions and also makes it potentially extensible in the future work.

A collection of photographs that meets the assumptions of a chosen model can be called a *model photographic collection*. If any of the suggested requirements or limitations are not

met by some real-world photographic collection, the estimated values of street segment attractiveness A_e become biased compared to what the chosen model is capable of. This happens because some geographic areas get redundant 'votes'. To avoid or at least to reduce this negative effect it is necessary to introduce an additional function B, which distorts the existing distribution of photographs by removing the images that do not meet the requirements. Alternatively, such function can assign numeric scores from 0 to 1 to all photographs in a given collection depending on some parameters, and this would allow a more reliable distribution of 'votes' to be obtained (assuming there are no major flaws in the model). Mapping function M_{cont}, which deals with a continuous geographical surface, can be replaced with some function M that works with a point-based collection of photographs C directly:

$$A_e = M(e, B(C)) \tag{1.2}$$

The value of gain $\mathbb{\Gamma}_u(P)$ that a pedestrian receives from walking through the attractive places is also individual, just as the degree to which he or she perceives the value of street attractiveness. Thus, a person's function $\mathbb{G}_u(\{\mathbb{A}_{e,u} : e \in P\})$ may not be accurately extracted. Assuming that people may have similar preferences, this function can be replaced with some function G, which would be applied to all pedestrians and would on average give a relatively similar result to individual $\mathbb{\Gamma}_u(P)$. Gain $\mathbb{\Gamma}_u(P)$ can be therefore replaced with $\Gamma(P)$. Because no studies were found about the possible nature of this function (i.e. how walking through a non-attractive street followed by a very attractive street compares to walking the same distance along streets with average attractiveness), in this work it will be assumed that $\Gamma(P)$ is a sum of the attractiveness scores of road segments:

$$\Gamma(P) = \Sigma(\{A_e : e \in P\})$$

Time or distance, which represent the cost of a walk ($\omega_u(P)$), can be linked with a straightforward linear relationship, as the speed of walking is not significantly affected by sharp turns, traffic congestion, speed limits or other factors introduced by the topology of a road network. Therefore, it is always possible to normalise individual $\omega_u(P)$ and ω'_u to some $\omega(P)$ and ω',

shared among all potential users of a routing system. In this work ω' is considered to be equal to the desired travel distance; if the available leisure walk budget is defined by a user as time, the value is converted to distance by its multiplication by the persons's planned average pace.

Thus, the task of deriving a good leisure walk P' from v_{start} to v_{end} through attractive places, given budget ω' and a crowd-sourced collection of photographs C, can be described as the following system of equations:

$$\begin{cases} \omega_u(P') - \omega'_u \geq 0 \\ P' = \underset{P(v_{start}, v_{end})}{\arg\max} \{\Sigma(M(e, B(C))) : e \in P(v_{start}, v_{end})\} \end{cases} \tag{1.3}$$

The implementation of a routing system, which would solve these equations, introduces the following questions:

Which collection of photographs C to use?

What function B to apply in order to reduce bias in C?

How to map the spatial distribution of photographs into the attractiveness scores for street segments (how to define function M)?

Which approach to choose for finding a route with the given cost ω' and a high value of gain Γ among all possible routes $P(v_{start}, v_{end})$?

The subjective nature of a measure of street attractiveness, which varies from an individual to an individual, makes it hard if not impossible to find ideal answers to the above questions, i.e. those that could be proved to outperform any other possibilities. However, it can be attempted to explore the opportunities and compare them to each other, thus moving towards the improvements in the problem of leisure pedestrian routing.

1.3 Research aims and objectives

This thesis attempts to develop a methodology for assessing user-generated geotagged photographic datasets for their use in planning leisure walks and also to propose a routing algorithm that can suggests attractive routes for pedestrians based on these data. Such aims imply two groups of objectives. The first one is purely related to the photographic collections and involves (1) the selection of candidate data sources, (2) their assessment and (3) processing (or filtering). The second group of objectives concerns combining photographic archives and the road network data. It involves (1) a study of how a point-based distribution can be mapped onto the edges of a routing graph and (2) the implementation of a routing algorithm that uses the combined data.

This research also looks at previous attempts to inform travel planning algorithms with the collections of photographs. This important part of the project helps identify the gaps that need to be filled in and also justifies how the study may contribute to the knowledge.

It was decided to limit the context of the work to urban areas, as they are richer on photographic data (Antoniou 2011) and have more dense road networks compared to rural areas.

Visual Analytics was selected as the key approach to the analysis. The reasons for using VA for the chosen task are explained in Section 3.3 on page 92.

1.4 Research questions

Research questions for this PhD study are determined by the questions introduced on page 22.

RQ 1: What sources of user-generated photographic data are suitable for automatic creation of attractive pedestrian routes?

A set of sources of georeferenced photographic data must chosen, photographic data must be collected and the analysis of patterns within the datasets must be performed. It is important to establish the characteristics of the data sources that make them suitable for using as an input in a routing system.

RQ 2: What features of geotagged photographs can be used in pathfinding?
The metadata of shared photographs must be analysed to determine whether or not they can be used for route generation.

RQ 3: What methods should be applied for data filtering in order to remove unwanted user entries?
If filtering of photographic data is required, it is necessary to establish methods for extracting unsuitable photographs and excluding them from the datasets.

RQ 4: How to obtain the attractiveness scores of the road network edges in a routing graph?
A study of ways of combining photographic data with the road network data must be conducted. Different mapping functions must be compared to each other, leading to some design recommendations.

RQ 5: How to implement the routing algorithm?
In order to be able to test the system as a whole, it is important to design and implement a sample algorithm for solving System of equations 1.3.

RQ 6: How to asses the results?
It is necessary to establish what methodology can be used for the evaluation of the work of the algorithm.

1.5 Thesis structure

The report starts with the review of existing work that is related to utilising photographic data in place assessment and trip planning. This part of the thesis helps understand what has already been done in the field and what gaps need to be covered. The next chapter considers the problem of photo-based leisure pedestrian routing in general terms and also justifies the methodology applied in this research. Chapter 4 focuses on a study of four chosen photographic archives and Chapter 5 moves towards combining of the results from Chapter 4 with the road network data. Finally, the closing chapter lists the outcomes of the research and discusses further steps that can be taken in the same direction.

Chapter 2

Review of existing work

2.1 Related research projects

There are two major ways in which volunteered geographic information such as geotagged photographs have been combined with trip planning, navigation or pathfinding. The first one suggests overlaying walking directions on top of the photographs with various landmarks, thus helping people navigate in the urban environment (Beeharee and Steed 2006; Hile et al. 2008, 2009). This approach involves landmark mining, image processing and 3D structure-and-motion reconstruction of the real world objects. The technology of overlaying travel directions over geotagged photographs has been patented in the United States (Herbst, McGrath and Borak 2008).

An exploratory trial held by Beeharee and Steed (2006) has revealed that if the walking instructions are supplemented with real photographs of places, they become easier to follow, thus making the navigation time shorter and reducing confusion. The above conclusion gave rise to at least one startup, which attempted to bring the idea to the market of mobile apps. 'Lumatic LLC' (`http://lumatic.com/`) opened in 2011 and released 'Lumatic City Maps' for iOS and Android. According to the description on company's website, this application provided "easy to follow, photo-based navigation for pedestrians and public transit riders"

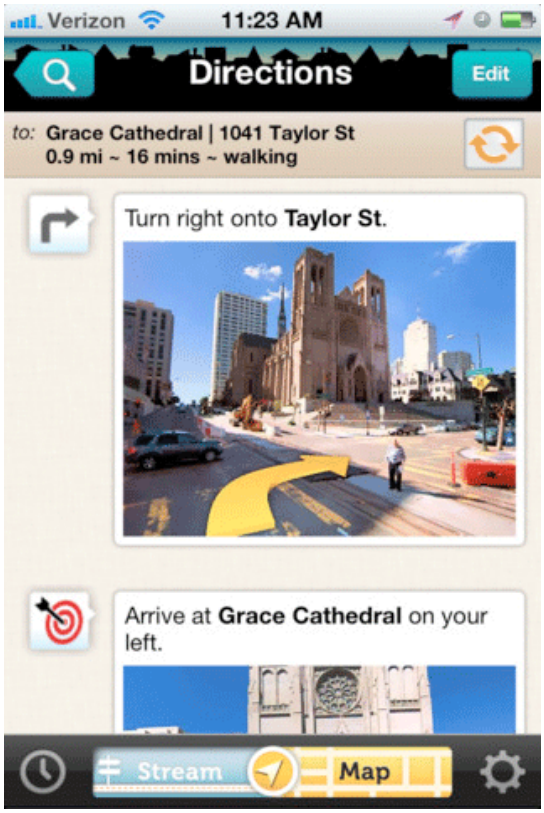


Figure 2.1: Example of landmark-based pedestrian navigation with use of photographic content. *Source: Lumatic City Maps,* `http://lumatic.com/`.

and could be used in San Francisco and New York City. An example of a landmark-based pedestrian navigation with use of photographic content is shown in Figure 2.1.

As of today, the above way of using crowd-sourced photographs in routing for humans is not very likely to to meet competition. A more structured approach to supporting pedestrians' turn decisions involves centrally produced 360° panoramas, which guarantee even coverage of streets and homogeneous quality of data. One such example is Google Street View, which is widely known in many countries (`https://www.google.com/maps/views/streetview`).

In spite of helping travellers with finding their way around a city, crowd-sourced photographs with overlayed instructions are not being involved in route generation. When somebody is willing to take a leisure walk along the attractive pathways using a similar system to the one described above, he or she can be only suggested to look at the photographs along a generated route, but the images themselves will not take part in the process of pathfinding.

The second way of linking geotagged photographic content with route and trip planning focuses on the extraction of a 'wisdom of the crowd' from the collaboratively generated large collections of images. This approach strongly relates to the topic of this research.

One of the first works aiming to build a system that can help tourists find attractive places using volunteered photographic information and thus plan their trips was that by Quack, Leibe and Van Gool (2008). This work "presented a fully unsupervised mining pipeline for community photo collections," an output of which was "a database of mined objects and events" that could be of interest for a tourist.

A similar research was conducted by Kisilevich et al. (2010). It was also not directly related to the pedestrian route planning and focused on the visualization of attractive places with use of density maps based on Flickr and Panoramio data. An example of such visualization can be found in Figure 2.2. This work also demonstrated the relationship between place attractiveness and the data available from the photo-sharing websites. Density-based spatial clustering together with visualization techniques allowed authors to estimate the main features of the detected attractive places by retrieving individual photographs with high influence weights from the derived clusters. According to the authors' belief, the proposed method could be "of great importance to travelers by reducing search time of attractive places, to providers of tourist services or researchers who analyze spatial events."

Figure 2.2: Clusters representing places of interest for tourists in Saint Martin island extracted from Flickr and Panoramio data. *Source: Kisilevich et al. (2010)*

Kurashima et al. (2012) focused on a problem of trip planning for tourists in more detail and proposed a travel route recommendation method that made use of the photographers' histories as held by Flickr. Recommendations were made based on a new *photographer behaviour model*, which estimated the probability of a photographer visiting a particular landmark. The concept consisted of two building blocks: the topic model, which estimated the user's own personal preferences, and the Markov model, which helped find typical routes that photographers took. The researchers assumed that the geotagged images could represent personal travel route histories, so sorted the locations indicated by the photographers according to their timestamps. An example of a route generated by Kurashima's system is shown in Figure 2.3.

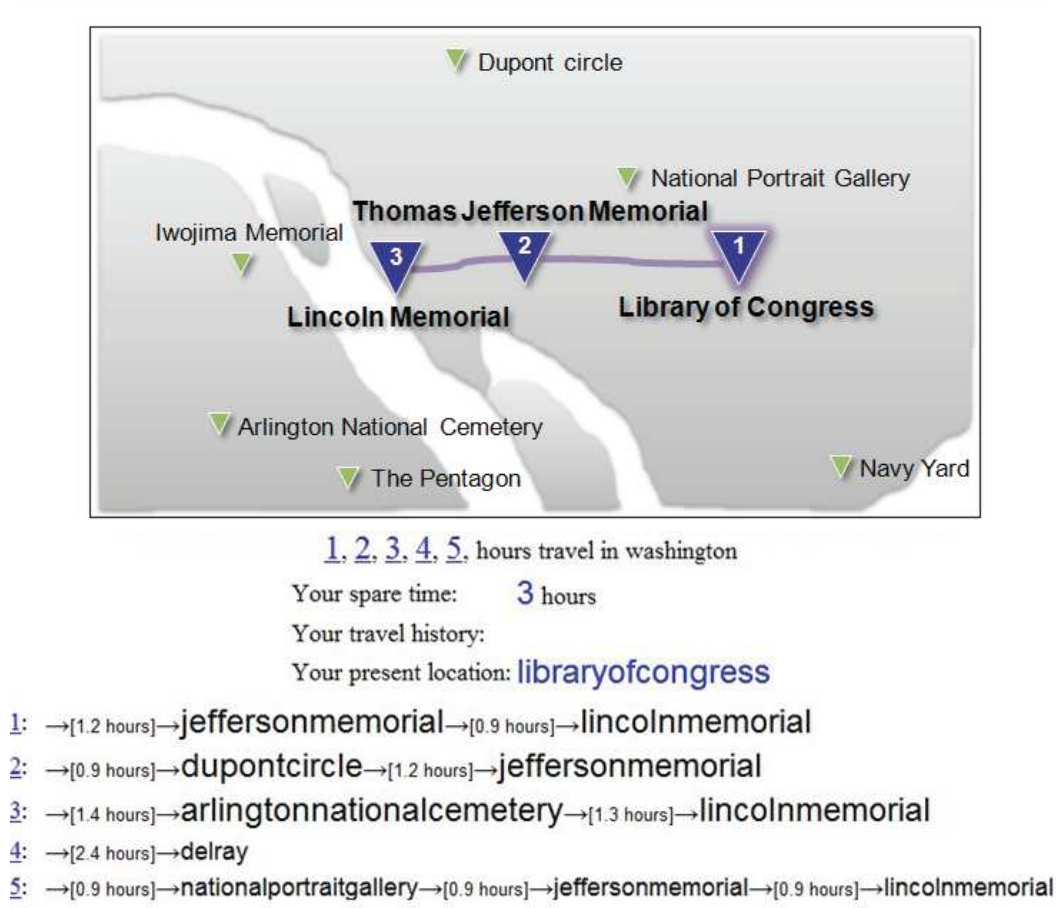


Figure 2.3: Example of a tour suggested by the travel route recommendation framework, which was designed by *Kurashima et al. (2010)*

Lu et al. (2010) from Microsoft also analysed personal photo streams and created a travel route planning system called Photo2Trip based on Flickr data. Using spatial and temporal attributes of crowd-sourced photographs, this system could suggest a customised trip plan for a tourist, i.e. a sequence of popular landmarks to visit and time to spend at each of the destinations. The core of the work was an internal path discovering algorithm that merged incomplete individual photo streams into combined paths that could form longer trips. As the research project mentioned recently, the system did not consider the street network, so the movements between the selected locations were shown as straight lines. The framework extracted travel and stay durations from the differences between the timestamps in the users' photo streams. An example of a route suggested by Photo2Trip can be found in Figure 2.4.

Similar systems were created by Okuyama and Yanai (2010) and De Choudhury et al. (2010, 2010*b*) – these projects also considered individual user photo streams to generate travel suggestions. Travel suggestions were designed to work as guides for tourists by telling them about what places of interest could be good for them to visit and how much time to spend at each location.

The idea of utilising individual photo streams for trip generation demonstrates promising results, but sets strict requirements to the underlying photographic data. If only a small proportion of photographers continuously take and share their works during the trips, the outcomes of the algorithms can be influenced by a personal bias, which, as previous research shows, does exist in real crowd-sourced photographic collections (Purves, Edwardes and Wood 2011).

(a) A 1.6 hours path (b) A 5-hour path

Figure 2.4: Output of Photo2Trip travel route suggesting system. *Source: Lu et al. (2010).*

The nature of the leisure *walks* is different. Having no long stops such as 'visiting a museum', a walk is a continuous movement in space. Time that needs to be spent for a walk has a linear dependency from the distance and the pace (speed) that are defined by a user. A problem of finding a path that does not exceed a given distance or time while maximising profit (in this case – the attractiveness of places that are passed by) is known as *orienteering problem with time windows* (Kantor and Rosenwein 1992). It is hard to be approximated efficiently, so the solution is approached with meta-heuristics algorithms such as genetic the algorithms or the ant colony optimization. Another option consists in reducing the problem to a classical shortest-path problem by using a weighted linear combination of all criteria as the cost function (Hochmair and Navratil 2008).

Antourage (Jain, Seufert and Bedathur 2010) is, perhaps, the first system that linked pathfinding with photographic datasets. This research approached the problem described above with the ant colony optimisation, in particular with a novel adaptation of the max-min ant system (MMAS) meta-heuristic. The locations of the photographs collected from Flickr were transformed into a vertex-weighted graph with a fixed inter-node distance, and thus the most attractive areas were distinguished. The trips in Antourage started from a specified location and visited popular places subject to a given distance constraint. An example of such tour is shown in Figure 2.5.

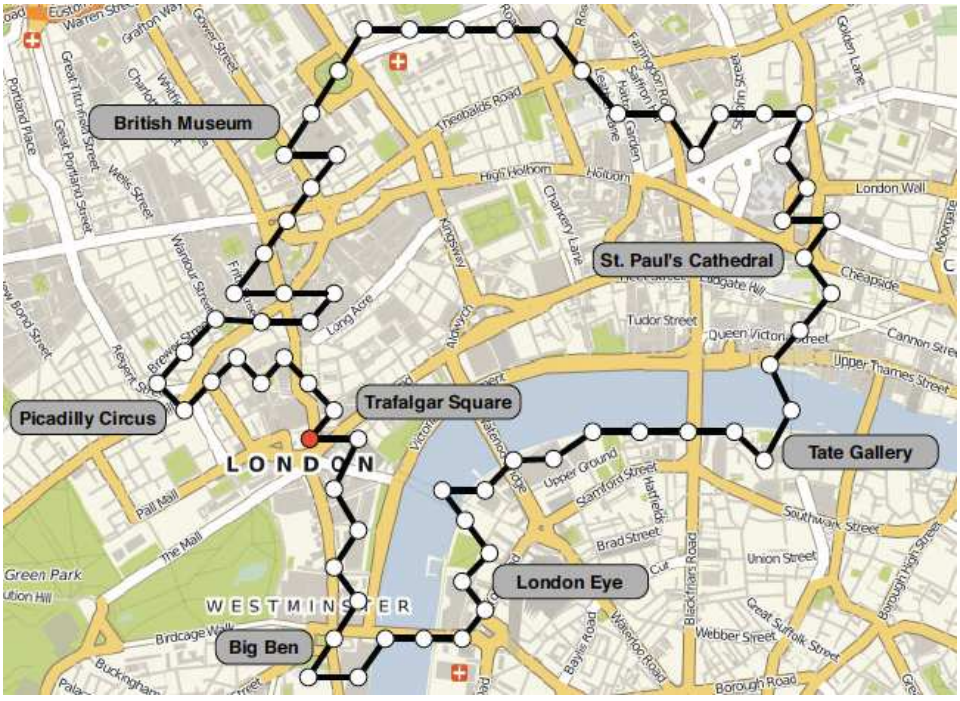


Figure 2.5: Example of a tour found by Antourage in London. *Source: Jain, Seufert and Bedathur (2010).*

Another interesting case study of linking photographic data with routing was recently conducted by Quercia, Schifanella and Aiello (2014). They divided the bounding box of Central London into cells of 200×200 meters and then collected subjective human opinion about each of the obtained 532 rectangular regions. The data was sourced from `http://urbangems.org/`, a website where volunteers ranked randomly shown locations as beautiful, quiet and happy in a form of a game. The photographs that people categorised were from Geograph and Google Street View. Based on these data and with use of Eppstein's routing algorithm (Eppstein 1998), it was possible to automatically suggest walkways that were not significantly longer than the shortest path between the two given locations, but could be perceived as more beautiful, quiet or happy than the default option. In addition to this outcome, the work featured an interesting approach to mixing gathered crowd-sourced subjective opinion with geotagged data from Flickr. It was found that the spatial density of Flickr photographs in combination with some of their attributes could give an estimate of some features of the urban environment.

The problem of generating attractive routes for pedestrians is similar to the same task for other modes of transport. Zheng et al. (2013) proposed a new method of mining *roadside points of interest* from crowd-sourced photographs. Their system, called *GPSView*, could discover a number of scenic roadways in northern part of California based on Flickr and Panoramio data. An interesting feature of the reported landmark-detection algorithm was its ability to estimate the visibility of points of interest (POIs) along the chosen route. This was done by means of matching the directions of the roads with the directions of *principle components* – vectors that were derived from the distributions of geotagged images around each of the detected POIs.

Another related research that considered the problem of extracting scenic driving routes was conducted by Alivand and Hochmair (2013). In line with the work mentioned recently, the experiments within this project also covered California and involved the same photo-sharing services (Flickr and Panoramio). The approach was based on an hypothesis that if a photographer "uploads several scenic pictures during a day trip that are located along a meaningful route (based on the time stamp and geometry)," one can "conclude that the traversed route is scenic". The result of this research is also a network of scenic routes for California.

2.2 Problems and scope for a new research

A review of the related research projects has shown that the idea of linking human navigation with the crowd-sourced geotagged photographic data has been rather topical in recent years and has been approached in a variety of ways. None of the discovered works, however, were suggesting to look at the problem of route finding as it was proposed earlier in Section 1.2 (page 17). A combination of these facts gave a positive general background for a new study.

It was observed that the importance of photographic data analysis in the related projects was likely to be underestimated. In spite of the fact that crowd-sourced images could contain photographs taken both during the day and over night, indoors and outdoors (Szummer and Picard 1998; Boutell and Luo 2004), during events (Andrienko et al. 2009; Rattenbury, Good and Naaman 2007), could be referenced incorrectly (Zielstra and Hochmair 2013) or simply be human portraits, relatively little attention was paid to the process of filtering of input data. For instance, Okuyama and Yanai (2010) only excluded the pairs of the photographs for which geotags were exactly identical but the time was different by more than five minutes; De Choudhury et al. (2010) filtered out the photographs with the time of photographing equal to the time of sharing plus those images that had no tags related to the locations where they were found at.

Some works attempted to obtain ‘clean’ distributions of ‘useful’ images by means of their filtering by textual attributes (e.g. Memon et al. 2014; Alivand and Hochmair 2013) or by working only with the contributions from the most active photographers (e.g. Lu et al. 2010; Zheng et al. 2013). Both methods are unlikely to guarantee the absence of various kinds of bias and even can potentially introduce more of it. The smaller the crowd of photographers and the size of the entire collection of records, the more vulnerable the results of pathfinding can become to personal bias (some supporting evidence of this statement can be found in Purves, Edwardes and Wood 2011). Textual attributes such as titles and tags may not be necessary explicit (Ames and Naaman 2007), written in one language and contain no typos.

It was also noted that none of the discovered related projects focused on the analysis of the image contents and some extra attributes such as EXIF tags (Camera & Imaging Products Association 2010). These features could be potentially involved in informing a route-suggesting system too.

Summarising the above, a detailed study of spatial, temporal and other attributes for a set of crowd-sourced photographic collections could generate some new knowledge at the intersection of two fields of information science – research of volunteered geographic information (VGI) and transport problem solving.

New research could also pay more attention to the process of harvesting spatial distributions of photographs. There exists evidence that the tools, which allow software developers to retrieve crowd-sourced collections of images from photo-sharing services, do not guarantee the integrity of the obtained data (Flickr 2007). "Most of the images cannot be easily found through common searches" (Hietanen, Athukorala and Salovaara 2011).

More ways of converting point-based geographic information to the numeric scores for road network edges could be considered in addition to the methods suggested by Hochmair and Navratil (2008). This could lead to some new insights about how the routing can be informed by photographic distributions and also encourage more experiments with similar data in future.

Finally, new research could suggest a generalised theoretical framework for utilising photographic content in transport-related problems. This piece of work, consisting of independent data-processing components, could be later reused by others in a variety of ways.

Chapter 3

Methodology

3.1 Research workflow

Considering the desired outcomes of a project as its starting point is crucial for the justification of the required actions. The main outcome of this research can be defined as *a set of recommendations and techniques to be used in an arbitrary routing system, which suggests attractive leisure walks based on crowd-sourced geotagged photographic content*. The developers of such systems will be able apply new knowledge and improve their services by understanding more about how shared geotagged images represent street attractiveness. Therefore, this research should introduce ways of overcoming potential challenges and issues.

Let's consider a general structure of an arbitrary routing system that uses crowd-sourced photographs. It is shown in Figure 3.1 on the following page and reflects the major processes and flows of data. All components of this system can be split into two categories: *data preparation* and *runtime*.

Data preparation is something invoked once or with a relatively low frequency. As in the case of a standard pedestrian routing system, it involves street network data gathering and topology graph building. To supplement this information with the scores of attractiveness, three additional steps are introduced: (a) retrieving the data from one or several photo-sharing websites, (b) transforming them and (c) assigning 'attractiveness scores' to all segments of the road network. After all these steps are complete, a system is ready to use.

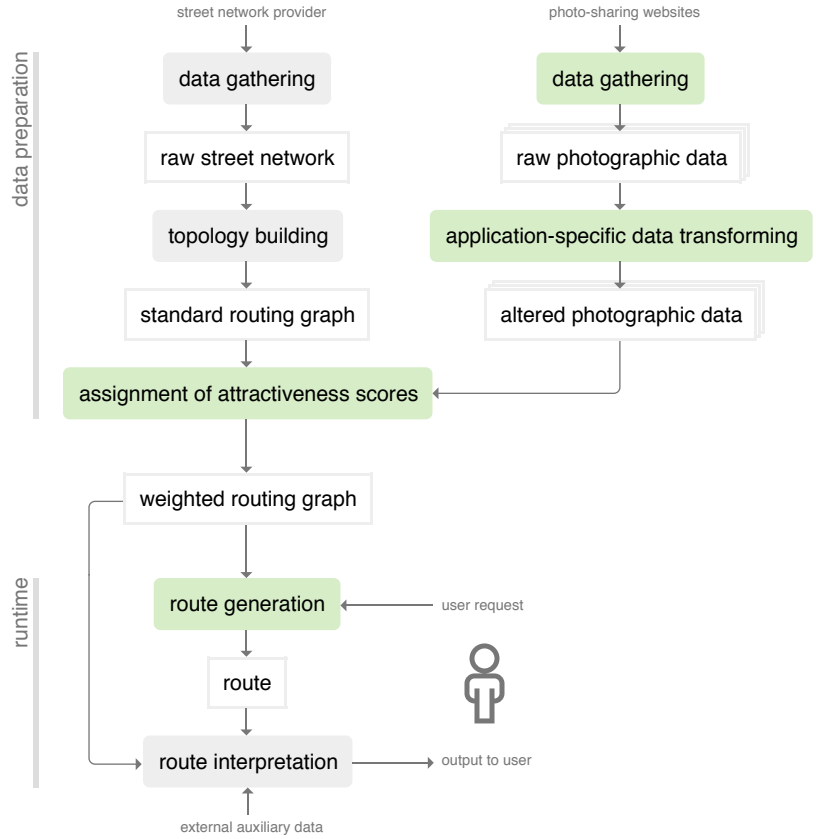


Figure 3.1: The structure of an arbitrary system that suggest attractive leisure walks using crowd-sourced photographic data. Highlighted are the components this research focuses on.

Runtime components are serving user requests. When somebody asks a system to generate an attractive route between the given locations, two units are involved to handle a response. First, a query is passed to a routing algorithm, which takes a weighted routing graph and finds one or several suitable routes that match the requirements and have high attractiveness scores. Because raw result cannot be consumed by a human directly, it is interpreted (visualized) by another unit, and the output of this process is passed to a user. The response can be supplemented with some external auxiliary data such as satellite imagery, nearby points of interest, map tiles, etc. to enrich the information a user consumes.

The components are not necessary limited to those listed above – a version of a photo-based routing system can work with other factors a pedestrian may consider when planning a leisure walk. For example, these can be a distribution of air pollution, land evaluation profiles or criminality rates. In order to keep the scope of the research clear, this work does not take any

additional factors into account and only focuses on street attractiveness extracted from the spatial distribution of photographs.

The quality of the walks that a photo-based routing system may suggest depends on the configurations of the components it consists of. By running experiments that lead to answer research questions defined on page 23, it is possible to determine and tune the processes, which are being involved, thus potentially moving towards a more accurate estimation of street attractiveness and a better route generating strategy.

Not all components of a photo-based routing system are within the scope of this research – the project does not involve any experiments with different sources of street topology as well as with route interpretation. The assessment of road network quality is a separate broad research topic; this project uses only a single source of street data and considers them a 'ground truth'. Route interpretation is also outside of the scope: this research does not examine how the obtained routes should be presented to a user and what auxiliary information they should be supplemented with.

The remaining components (highlighted in Figure 3.1) can be split into two categories according to the type of data they are dealing with, which will also agree to the groups of objectives in this research (page 23). An approach to selection, collection and transformation of the photographic data affects the amount of bias contained in a resulting approximation of street attractiveness. A set of experiments can help investigate how this bias could be reduced. Ways of combining photographic data with road network data as well as the approach to routing determine how the core system of equations (1.3 on page 22) is solved. It is possible to move towards an optimal (model) solution through the comparison of possibilities, and such a study forms a second set of experiments.

The contents of both sets of experiments were not known in advance; the justification of what was included in this research is described below.

3.1.1 Experiments with photographic datasets

The assumptions that were chosen for the experiments in this research (see Section 1.2, page 20), determine the following extended set of requirements to the whole collection of images and the items it contains:

1. A collection of images should only consist of photographs.
2. All photographs should be georeferenced (supplemented with geographical coordinates).
3. Attached georeferences should be valid (the photographs should be reported to be taken within a short distance from where they were actually made).
4. A collection of images should be formed by a reasonable number of photographers; the contributions made by individuals should be equivalent to eliminate personal bias.
5. Each photographer should take maximum one picture at any place, thus not 'voting' for the attractiveness of any area more than one time.
6. The size of a collection of photographs should be sufficient, so that the difference in popularity of streets among photographers is vivid (neighbourhoods of some walkways should have poor coverage, while the surroundings of others should contain hundreds of photographs, and this difference should be distinct).
7. All photographs should be taken outdoors.
8. All photographs should be taken at daytime (nighttime walks are not included into the scope of this research).
9. All photographs should be taken by pedestrians (there should be no aerial images or pictures from places that are unavailable to pedestrians).
10. All photographs should depict features of roads, which can be constantly found at the georeferenced location and may be of interest to others.

A *model photographic collection* (one that complies with all assumptions of a chosen photographers' activity model) would not require any transformations before it could be used for assigning scores to road segments. In other words, bias-reduction function B in formula 1.2 on page 21 for such a collection would be equal to an identity function.

If any of the given requirements are not satisfied by a collection of images, the produced spatial distribution of 'votes' may become distorted. This potentially makes the scores totally irrelevant or at least leads to a bias in the estimations of street attractiveness (compared to some theoretical best possible result that can be obtained with a given *model photographic collection*). For example, if a collection of images contains many indoor photographs, streets near popular buildings get higher values of attractiveness, and a routing algorithm gives them unreasonable preference. If, in some other case, significant quantities of photographs are georeferenced incorrectly, attractiveness scores become totally irrelevant.

It is important to assess any candidate source of images before it can be applied for estimating street attractiveness. If any inadequacies are found, it can be attempted to eliminate their impact. Irrelevant photographs (those that cannot be considered as 'votes' for street attractiveness) can be removed from a dataset, thus making it more similar to a *model photographic collection*. Some potential issues, however, cannot be resolved with this approach and therefore should lead to a rejection of a candidate source as such. For example, if the overall size of a photographic collection is small, no streets are able to get sufficiently high attractiveness scores, and this does not let a routing algorithm distinguish between attractive and non-attractive streets.

Because the attractiveness scores are calculated based on potentially large numbers of photographs, it is possible to suggest that not all 100% of images have to strictly meet all listed requirements in order for a dataset to be suitable for the chosen purpose. If there exists a small proportion of invalid 'votes' (e.g. some images are not photographs), bias is not introduced as long as there are no regularities in their distribution. Larger numbers of such cases increase the amount of noise in the resulting attractiveness scores, but do not make them systematically distorted. Of course, if the quantity of irrelevant images in a given potential source of street attractiveness is high, a collection of photographs cannot be considered as valid due to the unreliability of scores it can produce.

Instead of 'accepting' and 'rejecting' the photographs based on some filters (binary logic), it is also possible to assign numeric scores to all images according to their relevance to the estimation of street attractiveness (fuzzy logic). In this scenario, for example, a photograph taken inside of a restaurant, but showing a part of a nice street through a window, would also be considered as something contributing to the value of street attractiveness despite that requirement 7 is not met (an image should be taken outdoors). The rate of such a 'vote' would be made smaller than for one by a photograph of the same place taken outdoors. This approach looks interesting, but significantly increases the complexity of the bias-reduction function B – it makes it really hard to obtain adequate rates for all 'votes' (e.g. *What is a correct rate for a photograph that is taken during a sunset and contains a human face, which occupies 60% of the image?*). Due to the difficulties related to the verification of filters based on fuzzy logic, this research does not consider them as means to eliminate problems in potential estimators of street attractiveness.

When a bias-reduction function B is binary, the process of its application is more straightforward. A collection of images is given to a set of filters $F_1 \dots F_N$, and each of them assigns values of 1 or 0 (*pass* or *fail*) to the photographs according to some judgement rule. After the process is complete, function M, which maps 'votes' to street attractiveness scores (Equation 1.2 on page 21), only considers photographs that have passed all filters:

$$A_e = M(e, B(C))$$
$$B(C) = F_1(C) \cap F_2(C) \cap \dots \cap F_N(C)$$

All potential binary filters for a collection of images can be classified into two types: (a) those that make a decision based on some knowledge about the whole dataset and (b) those that reject or accept a photograph solely based on its own properties. An example of a *type a* filter can be a function that removes photographs taken by the same person at the same place (requirement 5); an example of a *type b* filter can be an algorithm that looks at the contents of a picture and rejects it if it is a drawing (requirement 1).

Because the method of combining binary filters follow the laws of binary logic, it is possible to optimise $B(C)$ by introducing the chaining of filters (Figure 3.2 on the facing page).

Filters of *type b* ($Fb_1, \ldots, Fb_m$ – those that consider properties of individual photographs) can be applied only to images that have successfully passed all filters of *type a* ($Fa_1, \ldots, Fa_n$ – those that take into account the whole collection of images). Such shift does not affect the contents of the resulting subset C_F, because $x_1 \cap x_2 \cap x_3 = x_1 \cap (x_2 \cap x_3)$. Besides, filters of *type b* can be ordered according to their cost starting with the least computation-intensive one, and this will allow even more resources to be saved. Such strategy is called 'lazy evaluation' and is often used by programming language compilers and interpreters (Watt 2006).

Taking into account that there are peculiarities in the behaviour of users that contribute to different photographic collections (Antoniou, Morley and Haklay 2010), there cannot exist a single universal bias-reduction function. Datasets from various origins may need diverse treatment in order to be made more similar to a *model photographic collection*. The same bias-reduction function, however, can be applied to distributions of images at multiple locations if they share the source, assuming that patterns in user behaviour do not significantly vary from city to city.

Specific binary filters may be found useful in one number of cases, while being ineffective or even inapplicable in others. Thus, the collections of images obtained from different sources require various combinations of filters, adjusted versions of the same filters or both.

The process of filter design and selection is closely related to the process of the analysis of photographic collections. It it necessary to consider each candidate photographic source separately to examine potential discrepancies with the requirements and to attempt to reduce them. This procedure is iterative and is shown in Figure 3.3 on the next page.

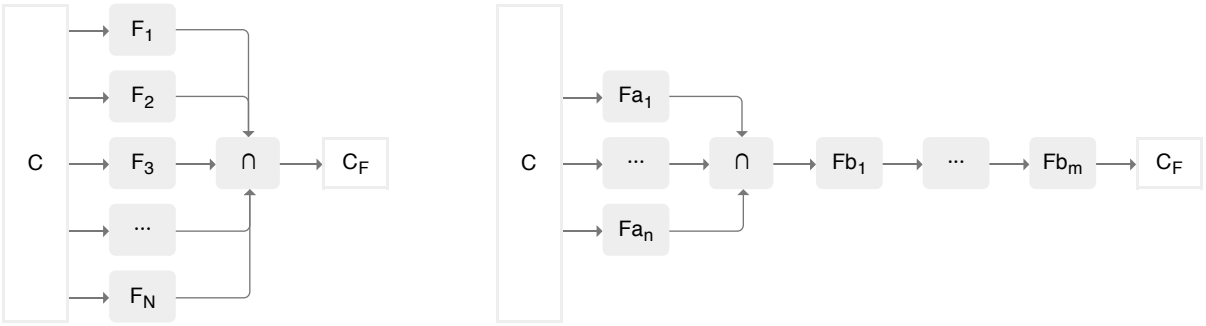


Figure 3.2: Optimisation of a binary bias-reduction function for a photographic collection using filter chaining.

Each iteration starts with the assessment of the dataset and it is attempted to find anomalies in the spatial distribution of images or signs of significant proportions of photographs that do not meet the requirements (page 38). If the discrepancies are unresolvable, the dataset is named inapplicable as a potential estimator of street attractiveness and is rejected. If the detected problems can be potentially eliminated, a filtering method is suggested and applied, which is followed by another round of data assessment. The process can be repeated for the same proposed filtering method if there are parameters that may influence the output. Several kinds of suggested filters follow each other until there are no more problems in data or actions to suggest. Finally, when (and if) the given dataset is clean, the procedure can be considered as successfully completed. Apart from a filtered version of the original collection of photographs, the result is a list of filters that have been successfully applied; this list describes the nature of the bias-reduction function. Having such information at one's disposal, it is possible to perform filtering for other collections of photographic data from the same source, assuming that general patterns in user behaviour do not differ across similar places or over time.

There may be cases when not all photographs, which do not meet the requirements, are excluded, however the dataset can be still considered as suitable. As mentioned earlier, such situation may occur if there no systematic character in the locations of misfiltered photographs. The presence of 'inappropriate' images in the spatial distribution of 'votes' would add noise to the values of attractiveness rather than introduce bias. Imperfections of the bias-reduction function can be acknowledged as 'assumptions'.

In this research it was chosen to consider a number of datasets from different popular photo-sharing services and to develop binary filtering methods which may help improve the estima-

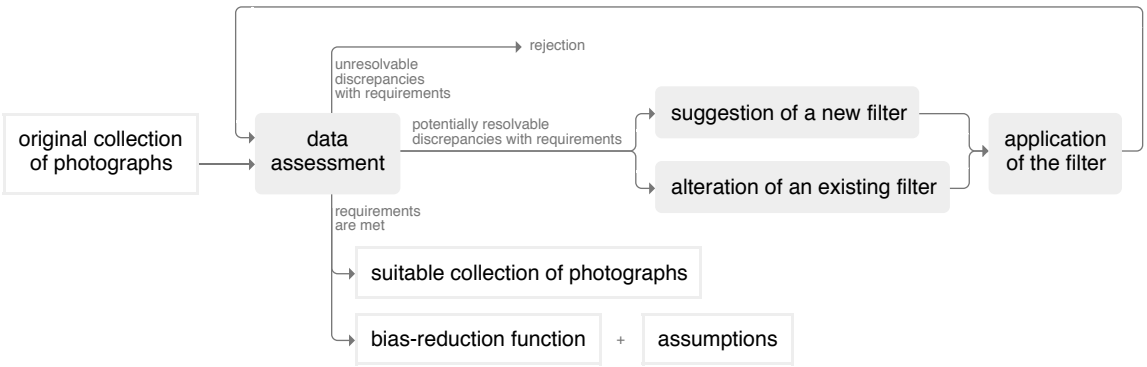


Figure 3.3: The procedure of photographic data analysis and filtering.

tion of street attractiveness with the given data. It was decided to perform the experiments in a single common geographic region, assuming that the same filtering rules may apply to other places due to similarity of patterns in user behaviour. The selection of a geographic region for the experiments is discussed later in Subsection 3.1.3 on page 57. It was chosen to treat all collections of data equally by adhering to the same workflow:

1. *Data gathering.* As photo-sharing services are often global, it is necessary to sample the existing data by only obtaining records from the chosen geographic region. It is not necessary to receive the files with images – attached metadata is sufficient at this stage.

2. *Detection of problems and anomalies in the distribution on photographs.* This step helps make general conclusions about the obtained dataset and check its compliance with requirements 2 to 6 (page 38). The dataset can be rejected at this stage in case if it contains unresolvable discrepancies.

3. *Suggestion of potential filtering methods, their application and verification.* The nature of detected anomalies in the distribution of photographs is examined, and the ways of their elimination are proposed and checked. According to the classification given earlier, the filters developed at this step are of *type a* (they take into account the entire collection of images rather than the individual instances).

4. *The analysis of the contents of the images.* This step aims to check requirements 1 and 7 to 10 and is reasonable only if the given collection of photographs passes previous steps without being rejected. It is necessary to check the validity of individual 'votes' to make sure that they are eligible for contributing to street attractiveness scores.

   Because is impossible to assess *all* images in a collection (there are tens or hundreds of thousands of items), it was proposed to conduct this study based on a random sample of photographs. This analysis generates the knowledge on what kinds photographs exist in the given collection and suggests whether it is possible to use the data straight away or further filtering is necessary.

5. *Suggestion of potential filtering methods, their application and verification.* The second round of filter development mostly concerns filters of *type b*, which aim to remove irrelevant images exclusively based on their contents and metadata. However, if the analysis

at the previous step reveals more problematic patterns in spatial distribution, more filters of *type a* can be proposed and tested.

6. *Conclusions about the quality the of data, the nature of the required bias-reduction function and assumptions.*

It is impossible to establish filtering rules before the investigation of the datasets, as there is a strong relationship between the nature of the detected problems and the required bias-reduction function. However, it can be attempted to highlight some potential means that could be used. Their variety is naturally limited by the data available about each item in a collection of photographs. In the best case scenario these are: (1) an image itself, (2) metadata generated by a camera (3) metadata either manually contributed by an owner of a photograph or added by other online users, (4) metadata from the moderators of a photo-sharing service.

The most important properties of images at the early steps of the workflow are the *spatiotemporal coordinates*, which are either automatically assigned by a camera or manually added by a photographer. Problems in these data may significantly affect the applicability of the given distribution of images, but are the most easy to investigate. Taking only these properties into account, it may be possible to make general conclusions about the considered photographic source and reject it if any coordinate-related problems are unresolvable. Spatial and temporal coordinates may be used in filters of *type a* at step 3 to potentially improve the dataset's compliance with requirements 2, 3, 4, 5, 6, 8 and 9.

Metadata either generated by a camera, a photographer, an online user or a moderator is a set of textual or numeric properties that describe the contents of a photograph. The list of available characteristics may vary for different sources of photographic collections, and not all values can be considered as objective and trustworthy. It can be attempted to use metadata for the development of filters of *type b* if the analysis of the contents of the images confirms the existence of large proportions of 'invalid votes' in a dataset (requirements 1, 7, 8, 9 and 10).

The images are the most reliable information to be used for finding 'invalid votes' (filters of *type b*). However, the involvement of these data in filtering is also the most resource-intensive. As of time of writing (2014), there are no automated algorithms that would help

obtain a comprehensive description of a scene by a bitmap, and manual classification of all images in a potentially large collection of photographs appears to be too expensive. Image analysis also implies working with significantly larger volumes of data compared to textual attributes. It can be attempted to use images for bias reduction if a study at step 4 suggests that trying this approach is beneficial. Image files can potentially help in dealing with requirements 1, 7, 8, 9 and 10.

Because of a tight relationship between data analysis and processing and also due to a subjective nature of the problem, it was decided to use a visual analytic approach as a key method for dealing with photographic collections. The details of the approach are described in Section 3.3 on page 92.

It was chosen to conduct the study of the contents of photographs (step 4) in a form of a publicly available survey. Similar approach was also used by Hochmair and Zielstra (2012) when analysing accuracy of geotagged photographs and by mySociety (2009) to measure place 'scenicness'. It was suggested to randomly select 300 images from the datasets that successfully passed steps 1, 2 and 3 and to ask the participants classify them according to a set of predefined criteria. The reason for involving more than one person in photo content assessment was a desire to obtain more reliable results. The quality of information was crucial as it would be considered as 'ground truth' and allow important statements about the photographic collections to be made. Besides, the obtained knowledge was also forming a basement for the development and verification of filters at step 5.

As the number of survey subjects (i.e. photographs) was meant to be rather large, it was decided to randomly assign queues of images to all new users and let them assess the given items one by one for as long as they wanted. On one hand, this approach could facilitate a smooth coverage of subjects with the responses, and on the other hand, it reduced the proportions of insincere responses, as people were given a right to stop the process when they wanted to. Even a small contribution, such as the assessment of two or three photographs, was adding to the results of the study.

It was important to keep the survey simple, as complex questions, which require extra cognitive load, could reduce user engagement and thus lead to a smaller response rate (Deutskens

et al. 2004; SurveyMonkey 2014). It was chosen to include only close-ended questions and limit the space of possible answers to minimum: *yes*, *no* and *hard to say* in most cases. The latter choice was introduced to reduce the number of false responses – if a photograph could not be easily classified by some criteria, without this option the participants could become confused and either randomly choose between *yes* and *no* or even quit the survey.

The following questions were assigned to each photograph in a sample:

1. *Is the image a real photograph?*
   This question was to check if an image met requirement 1 (see page 38).

2. *Is it a photograph of something outdoors?*
   This question was to check if an image met requirement 7.

3. *At what time of the day is the photograph taken?*
   This question was to check if an image met requirement 8. Unlike all other questions, the set of answers consisted of four choices: *day, night, twilight, hard to say*.

4. *Is it a photograph of something temporary?*
   This question was to check if an image partially met requirement 10. It helped detect photographs that were either taken during special events or mainly contained moving objects (cars, animals, etc.).

5. *Are people the main subject of the photograph?*
   This question was to check if an image partially met requirement 10. Portraits (photographs with one or several humans as the main subject) cannot be considered as valid 'votes' and therefore an attempt to detect and filter them had to be made.

6. *Could the photograph be taken by a pedestrian?*
   This question was to check if an image met requirement 9.

7. *Does the photograph suggest this is a nice place to walk?*
   The respondents were asked to leave their subjective opinion about a place they saw in a photograph. It was expected that the images, which passed all formal requirements, were indeed valid 'votes' for street attractiveness. If most of the photographs in one or more collections complied with the requirements, but people still reported that there

were very few images showing attractive walkways, either something was wrong with a source of photographs or with the chosen approach in general.

The first sketch of the user interface for the survey and its final design are shown in Figure 3.4 on the following page. Initially, apart from the classifications mentioned above it was also intended to include a question about the accuracy of the photographs' geographic location. This question and a map were removed after a round of discussions with colleagues. It was agreed that spatial accuracy could only be assessed by local experts, who could not be exclusively recruited for an online survey available worldwide.

A screen (a web page) in the survey was dedicated to one photograph at a time, which was shown on the left. Answering was performed by moving corresponding switches to required positions either with a keyboard, a mouse or a finger tap. Below was the 'next' button, followed by a representation of a random queue of photographs, showing the current progress. The queue was extended once its end was reached. In case if an image was reported as not a photograph (i.e. the answer to question 1 was *no*), all further questions became disabled as irrelevant to save respondents' time and avoid confusion. Similar rule applied to question 2: if a photograph was considered as taken indoors, questions 3, 4, 6 and 7 were not asked. A response could be only submitted after all available classifications were made. It was instantly added to the database, so a user could quit at any time with no need to save their answers.

The approach to survey data analysis is described in Subsection 3.3.4 on page 101.

The correctness of the proposed filters as well as the validity of general conclusions about image sources do not only depend on the steps taken during the analysis. It is also important to make sure that the datasets, considered in this study, reflect real distributions of photographs at their origin. If some data are systematically lost during gathering (i.e. at step 1 on page 43), photo collection C will contain extra bias, which may require additional filters as parts of function B or can even make a dataset totally incompatible with the requirements. This situation may happen when there is no direct access to the source database with all user-generated data, and the information needs to be collected with the use of an API (application programming interface). To avoid or at least to detect this problem, it is necessary to study the limitations of the APIs and make sure that they are used appropriately. Problems in data gathering (if any)

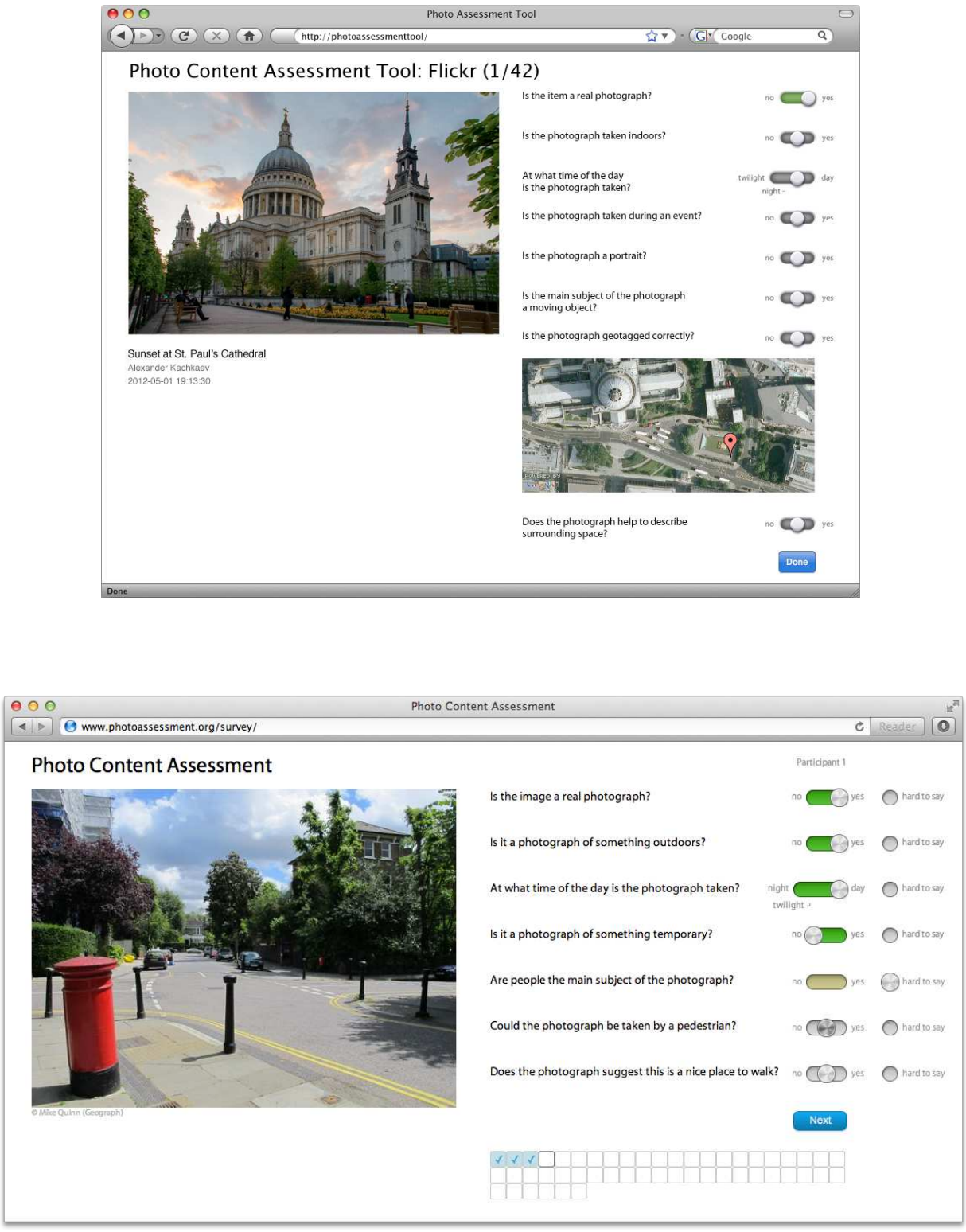


Figure 3.4: Interface of the photo content assessment survey. *Top:* initial sketch. *Bottom:* implemented version available at `www.photoassessment.org`.

should be reported separately from the problems in the distributions as such, as the first ones can be solved over time, and the latter ones have a permanent nature. A short study of photo service APIs was added to the workflow as part of step 1.

Suggested methodology leads to answers to research questions 1, 2 and 3 (see page 23). New knowledge about the collections of images from popular photo-sharing services can better explain their potential and limitations in estimating street attractiveness. This may help the developers of photo-based routing systems in future.

3.1.2 Experiments with the street network and routing

The second part of this research considers two other components of an arbitrary photo-based pedestrian routing system: *assignment of attractiveness scores* and *route generation* (see Figure 3.1 on page 36). These components deal with a road network graph, which consists of geographically distributed nodes (vertices) V and edges E. Every edge is characterised with some weight ω_e (distance) and an estimate of gain A_e (attractiveness). The edges are not necessary straight and may contain turns, but they cannot cross themselves or include any junctions.

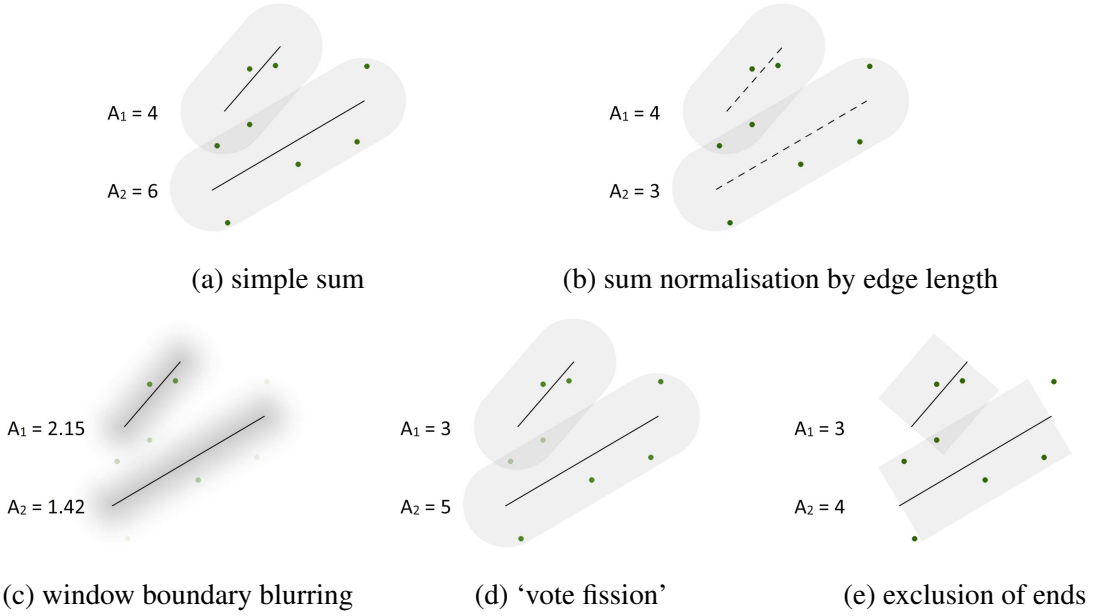
The subject for the first set of experiments is function M that converts the spatial distribution of photographs into the edge attractiveness scores (Equation 1.2 on page 21):

$$A_e = M(e, B(C))$$

To assign attractiveness score A_e to edge e, function M should consider a distribution of valid neighbouring 'votes' (photographs that have successfully passed through a bias-reduction function B) and return a numeric value based on their quantity and arrangement. The bigger the value, the higher the gain of walking through road segment e.

The core of function M is a window, within which the 'votes' constitute the estimation of attractiveness. The boundaries of a window may be either sharp or smooth, and there is an endless variety of forms and sizes that can be chosen to form its shape. For this research it was decided to pick a limited set of possible window designs and compare attractiveness scores A_e that they generate.

If bias-reduction function $B(C)$ is binary, all photographs, which have successfully passed through the filters, are considered to have equal importance. However, their contribution to the attractiveness score may differ depending on their remoteness from the road network edge or other factors. The simplest window design (Figure 3.5a) barely counts photographs located within a given range of meters from an edge. It does not consider relative positioning of edges, their lengths or other features. The second window design (Figure 3.5b) attempts to normalise the scores by dividing them by edge lengths. This transformation may be useful if the lengths of edges in a routing graph are too diverse or when a pathfinding algorithm needs to consider relative estimates of attractiveness. If experiments with different window sizes demonstrate that they should be rather large, it may be reasonable to suggest that more remote 'votes' can be given less weight, and thus a window can be made blurred (Figure 3.5c). The type of blurring (linear, normal, etc.) is a subject for discussion. It is also interesting whether it is 'fair' to count the same 'vote' more than one time if it fits into multiple windows. 'Vote fission' can be tried to test this aspect (Figure 3.5d). End exclusion is the final effect to consider (Figure 3.5e). This approach to window shaping can be potentially useful near joints between edges as it does not allow some 'votes' to be counted more than once similarly to the previous method.



(a) simple sum (b) sum normalisation by edge length

(c) window boundary blurring (d) 'vote fission' (e) exclusion of ends

Figure 3.5: Window designs for the attractiveness mapping function.

The second set of experiments in this part of the research is related to pathfinding and route evaluation. This project is not aiming to propose the most optimal high-performance algorithm for route generation, as this would excessively broaden the scope of the research. Any method of solving System of equations 1.3 on page 22 (which provides a definition of a good attractive walk) suits the goals of this work and allow all research questions to be answered.

In a common scenario, routing in a transport network is approached with one of the algorithms that solve a *single-source shortest path problem (SP problem)*. These are Dijkstra's algorithm, Bellman–Ford algorithm, Floyd's algorithm, A* and others. (McHugh 1990; Lerner et al. 2009; Zeng and Church 2009). Dijkstra's algorithm (Dijkstra 1959), the most straightforward solution for finding a path with the lowest cost given origin and destination, scans all possible moves from the source node to other nodes until it reaches the target. This approach guarantees that the obtained solution is always optimal, but requires significant amounts of computational effort when origin and destination are far from each other and a graph contains large numbers of connections. Some shortest-path algorithms attempt to decrease the number of operations required to reach the destination by reducing the number of verticies (nodes) that need to be visited until the searched node is found. For example, A* (Hart, Nilsson and Raphael 1968) first looks for a solution by scanning only nodes between the origin and destination, and if any obstacles are found, the search is broadened. This method works if the layout of a graph is known, which is always true for transport networks when edge costs are distances between the nodes.

When there is more than one factor forming an optimal route, pathfinding problem can be sometimes reduced to a SP problem by bringing various characteristics of the edges to a weighted sum of costs:

$$\omega_e = max\{0, \sum_{f=1}^{F-1} \omega_{f,e} \times I_f\} \tag{3.1}$$

Where:

ω_e is the weighted sum of all costs for edge e,
F is the number of factors,
I_f is importance of factor f (has to be non-negative),
$\omega_{f,e}$ is the value of factor f for edge e.

This approach has been used by Hochmair and Navratil (2008) to find scenic routes based on the proximity to parks, rivers and other attractive places. The idea is in converting the gain of selecting roads near scenic features into the reduced costs of corresponding edges. This distorts the graph and makes a 'shortest path' longer in terms of its real distance, but more beneficial in terms of the total value of the overall collected gain. Weights $\omega_{f,e}$ for gain factors are expected to be negative, so the higher their absolute values, the 'cheaper' a given preferable road segment becomes. Importantly, the weighted cost of any segment cannot be made less than zero, because this may create loops with negative total weighted costs, and it will become impossible to find a solution to the SP problem.

Converting a routing problem into a SP problem may help obtain attractive leisure walks with predefined time budget, but this approach has two limitations:

It is not possible to compute the required coefficient for gain knowing the desired value for the optimal route's real cost (i.e. the budget of a walk). The balance between the real cost of a route (its distance) and the gain (its attractiveness) entirely depends on the configuration of the road network. This information becomes lost when the weighted edge cost is calculated and can be only restored after an optimal route has been found. Thus, it is only possible to guess the best values of coefficients I_f by considering multiple cases iteratively. Once the real cost of the obtained optimal path has matched the given budget, the task can be considered as solved.

The increase of importance factor for gain does not necessary increase the real cost of an optimal path. There may be situations when further reductions of the weighted cost do not change an optimal route or even make it shorter. Thus, if the given budget has not been reached after the round of guesses, pathfinding problem becomes unsolvable.

For example, this can happen when there is a link between the origin and destination that has the highest values of gain in the entire network. Once the network is distorted enough for this route to become the most optimal, no other choices are able to 'compete' with it as further reductions in their cost with increase of I_f for the attractiveness factor also result the reduction of costs in this particular route. If the pathfinding problem suggests that nearly all given budget needs to be spent, the solution can be only reached by forcing the exclusion of some segments from an optimal route, which steadily forms a non-optimal solution. In other words, a *penalty list* with edges needs to be introduced.

The above limitations can be bypassed by choosing alternative approaches to pathfinding such as those that imply artificial intelligence. Examples are the solutions based on an ant-colony algorithm (Jain, Seufert and Bedathur 2010) or a genetic algorithm (Pahlavani, Samadzadegan and Delavar 2006). These methods can potentially demonstrate higher performance in large-scale road networks, but require more resources for their implementation and testing. As the goal of this research is not to propose the least computation-intensive algorithm that would solve System of equations 1.3 on page 22 in the most optimal way, it was chosen to involve a modification of the approach by Hochmair and Navratil (2008) to run the experiments.

The algorithm for planning an attractive leisure route between nodes v_{start} and v_{end} given distance budget L' in this research is the following:

1. Find the shortest path between the given nodes. If routing is not possible or if its distance L is larger than budget L', report that the solution does not exist for the given input.
2. Take a set of importance coefficients I_a for the street attractiveness factor and find 'shortest paths' for each of them (I_a is a special case of I_f).
3. Compute distance L and and gain G for all obtained routes.
4. If L of any of the given routes matches L', the solution is found. Otherwise:
    4.1. Among the suggested 'shortest paths', choose one with the smallest positive $L' - L$. Consider this route and a corresponding coefficient I_a as a base coefficient ($I_{a\,base}$).
    4.2. Find an edge in the base 'shortest path' that has the smallest gain-to-length ratio.

4.3. Add this edge to the *penalty list*.

4.4. Run SP algorithm again using $I_{a\,base}$ and assuming that edges from the *penalty list* have extremely high weighted costs.

4.5. If the distance of the resulting route is still less than L', return to step 4.1. Otherwise, report the latest obtained route as a solution.

The following modified formula was proposed for computing weighted edge costs ω_e:

$$\omega_e = l_e(1 - R \times min\{1, I_a \times \frac{A_e}{A_{mean}}\}) + penalty_e \tag{3.2}$$

Where:

l_e is the length of edge e (i.e. distance – its real cost),

R is the maximum ratio, at which attractiveness can influence the weighted cost of edges,

I_a is the currently used importance coefficient for attractiveness,

A_e is the estimate of attractiveness for edge e,

A_{mean} is the average value of attractiveness in the given road network,

$penalty_e$ is a large numerical constant for edges in a penalty list; zero for all other edges.

According to the formula, weighted cost of an edge is always in range between $l_e \times (1 - R)$ and l_e, so if $0 < R < 1$, the minimum possible value for ω_e is a small fraction of l_e. Thus, the weighted cost of any edge is always positive. R defines the maximum impact of attractiveness on ω_e: the bigger the value, the more distorted a routing graph (and therefore a 'shortest path') can potentially become. The values of A_e for streets with very similar attractiveness can significantly differ if the sizes of underlying photographic collections are too diverse. Therefore, the normalisation of attractiveness scores is introduced ($\frac{A_e}{A_{mean}}$). Both A_e and A_{mean} are positive despite that the attractiveness is a 'gain' and weights $\omega_{f,e}$ for gain factors are expected to be negative. This contradiction is resolved with a minus in front of $R \times \cdots$. Mean value of attractiveness is used instead of a median to avoid a possible zero in the denominator – this can happen with a median if more than a half of existing road edges have no 'votes' (e.g. when a chosen region is not very attractive or when a used photographic collection is relatively small). The normalised attractiveness score is multiplied by I_a, so the higher the chosen importance

coefficient, the smaller the weighted costs of attractive edges become. When I_a is equal to 1, all edges with $A_e \geq A_{mean}$ get $\omega_e = l_e \times (1 - R)$. Thus, they are considered equally preferable and are very likely to be included into a 'shortest path' if $0.9 \lessapprox R \lessapprox 1$. As it was mentioned earlier, the effect of increasing I_a starts eliminating after reaching some adequate limit. If its value is extremely high, edges with any positive A_e make $\omega_e = l_e \times (1 - R)$, which at some point starts meaning that *walking along a street with one 'vote' is equally beneficial as choosing an edge with a thousand of votes*. This negative consequence must be taken into account when defining a set of values for I_a at step 2 of the routing algorithm. Summand $penalty_e$ is used to help SP solver avoid unwanted edges. Unlike the exclusion of street segments from a road network, added penalty cost keeps the routing graph always connected. Thus, if all paths from v_{start} to v_{end} go via the given edge e, its 'penalising' will not break the link, and 'shortest paths' will still contain e.

It was chosen to use Dijkstra's algorithm to solve SP problems as parts of the proposed approach to pathfinding. This algorithm is straightforward and broadly applied, so it is likely that there exist well-tested robust implementations, which could be utilised. Even if performance of the resulting route-generating component is low, and attractive walks are not suggested in real time (i.e. in less than one or two seconds), the software would be still sufficient to test the routing system as a whole and to answer the research questions. Because Dijkstra's algorithm is always scanning topology graphs in all directions until the destination node is reached, it was decided not to work with large geographic regions and test all suggested methods in an urban area of a few square kilometers.

Another limitation of the chosen routing algorithm to be mentioned is its inability to suggest circular routes, i.e. handle cases when $v_{start} = v_{end}$. This can be a problem in a real walk-planning system, as such feature may be in demand. The following workaround can be applied to the algorithm suggested above:

1. Randomly pick a node $\dot{v}_{start}$ in the neighbourhood of v_{start}. If possible, give preference to nodes surrounded by edges with higher values of A_e.
2. Solve SP problem for v_{start}, $\dot{v}_{start}$ and $I = 0$.
3. Add all edges in the resulting route to the 'penalty list'.

4. Extract and save the length of this route ($\dot{L}$).
5. Find an attractive leisure walk with budget $L - \dot{L}$ between $\dot{v}_{start}$ and v_{end}.
6. Combine this route with the route that was obtained in step 2.

Circular routes were not included in this work's experiments.

Although other components of a routing system such as *road network data gathering*, *topology building* and *route interpretation* are outside of the scope of this research, it is necessary to have some basic implementation of them in order to run the experiments.

It was decided to apply a publicly available road network form a single source and use existing software to convert it into a topology graph. No data assessment or processing were included in the workflow except for the minimum required verification of the graph validity. In order for a routing algorithm to be able to find a path between any two nodes, it is necessary to check the connectivity of the graph and either fix errors in data or remove disconnected fragments of the network. Even if the accuracy of the street locations is not ideal, there are links via inaccessible areas or some paths are missing, the experiments with all suggested methods can be still made. In real routing systems the quality of the underlying street network can be either ensured by the data provider or improved over time with the help of feedback from customers.

Interpretation of the results to users may affect trustworthiness of the suggested routes and also forms a general impression about the service. Therefore, in a real routing system this component must be designed very carefully. As this research aims to benefit software developers rather than the end users, it was decided to make the functionality of the route interpretation component minimal, simply enough to see the results of conducted experiments. QGIS was chosen as a platform for this purpose; the details are described in Subsection 3.3.1 on page 96.

Once the whole photo-based routing system is implemented, it is important to evaluate the paths it suggests. The best option would be to run a user study, which would involve real people having real walks as suggested by the algorithm. As this work does not aim to provide a ready-to-use consumer product and also focuses on the visual analytics as the key research method, it was decided to limit the evaluation of the algorithm with a set of sensitivity experiments.

The objective of these experiments is to find out how the changes in the algorithm's configuration affect the results it can produce. The importance of bias-reduction (filtering) can be demonstrated with the same approach. It was chosen to conduct sensitivity analyses for (1) ratio R in Equation 3.2 on page 54, (2) existence of bias-reduction (function B) as a whole and (3) a single chosen filter in B. Multiple importance coefficients I_a are tried in each experiment (start at 0, step by 0.1).

The suggested sequence of experiments and methods leads to answers to research questions 4, 5 and 6 (see page 23).

3.1.3 Selection of a geographic region for experiments

According to earlier discussions in Subsection 3.1.1, all experiments in this research, related to photographic datasets, can be conducted in a single geographic region. In addition, the design of the routing algorithm (Subsection 3.1.2) suggests not to work with large-scale street networks, as time-complexity of the involved Dijkstra's method depends on the size of a routing graph (Aho, Hopcroft and Ullman 1974). Thus, it has been found rational to consider a single urban area as a place for all experiments, and London was chosen for the following reasons:

- This is the largest European capital (EuroStat 2014) and a top-visited city in the world (Forbes 2014).
- Interest in leisure walking in London has increased over years (WalkLondon 2012).
- There has been a number of studies of crowd-sourced photographic data in areas that include this city (e.g. Jain, Seufert and Bedathur 2010; Lee, Greene and Cunningham 2011; Antoniou 2011).
- Author's local knowledge can help analyse spatiotemporal patterns in data.

It was decided to pick an area of about 100–200 km^2, which would cover central parts of the city (approximately Travel zones 1 and 2). Despite that such region is relatively small

compared to the size of the whole Greater London (≈1,500 km^2), it has a rich variety of places, from busy tourist attractions, embankments and parks to quiet residential areas and industrial zones. This diversity suggests that the majority of possible patterns in user behaviour, existing in any considered photo-sharing service, are likely to be found within this compact region.

Before setting the exact spatial boundaries of the area for experiments, it was necessary to choose the shape of the region and a coordinate reference system.

As this research had no intention to produce a customer facing routing system, it was not necessary to make geographical boundaries meaningful, i.e. include or exclude some administrative or physically existing territories in full. Hence, preference was given to technical simplicity, and it was decided to work with a rectangular boundary, all sides of which are aligned to a coordinate grid.

Two coordinate reference systems were considered as candidates: World Geodetic System 84 (NGA 2004) and Ordnance Survey National Grid 36 (Harley 1976; Ordnance Survey 2012). The differences between these options are shown in Table 3.1:

	OSGB36	WGS84
formats	easting, northing grid cell, easting, northing	latitude, longitude latitude, longitude (+ minutes & seconds)
examples	530269, 179638 TQ 30269 79638	51.500119, –0.124614 51°30′02.43″N 000°07′28.61″W
coverage	United Kingdom	global
projection on a plane	straightforward, no transformations are needed	requires reprojection in order to keep proportions of objects
calculation of distance and area	easy, as each grid cell is 1×1 m²	resource-intensive, as the process involves complex transformations
usage	Ordinance Survey maps, local GIS projects	GPS, global mapping projects (OpenStreetMap, Google Maps, etc.), popular photo-sharing websites

Table 3.1: Comparison of WGS84 and OSGB36.

Despite that OSGB36 was easier to work with, preference was given to WGS84. As this coordinate system is global, all software, implemented during the experiments, can be reused in other urban areas, including those that are outside the UK. Besides, because WGS84 is one of the today's most common geodetic standards (Briney 2008), it is likely to be supported by

the majority of existing programs and libraries. These tools can be utilised in this project as standalone software or as the components of the routing system. The formulas for calculating real distances based on coordinates in WGS84 can be found in Movable Type Scripts (2012).

Direct projection of WGS84 on a plane (such a screen or paper) results significant distortions in local geometries, making objects in most parts of the world vertically or horizontally stretched. For this reason, all maps shown in this work are reprojected to EPSG:3857, unless otherwise stated. EPSG:3857 (commonly referred as Pseudo Mercator or Web Mercator) is a Spherical Mercator projection coordinate system, commonly used in mapping and visualization applications (EPSG Registry 2014). It keeps the proportions in local geometries and is supported by the major providers of map tiles.

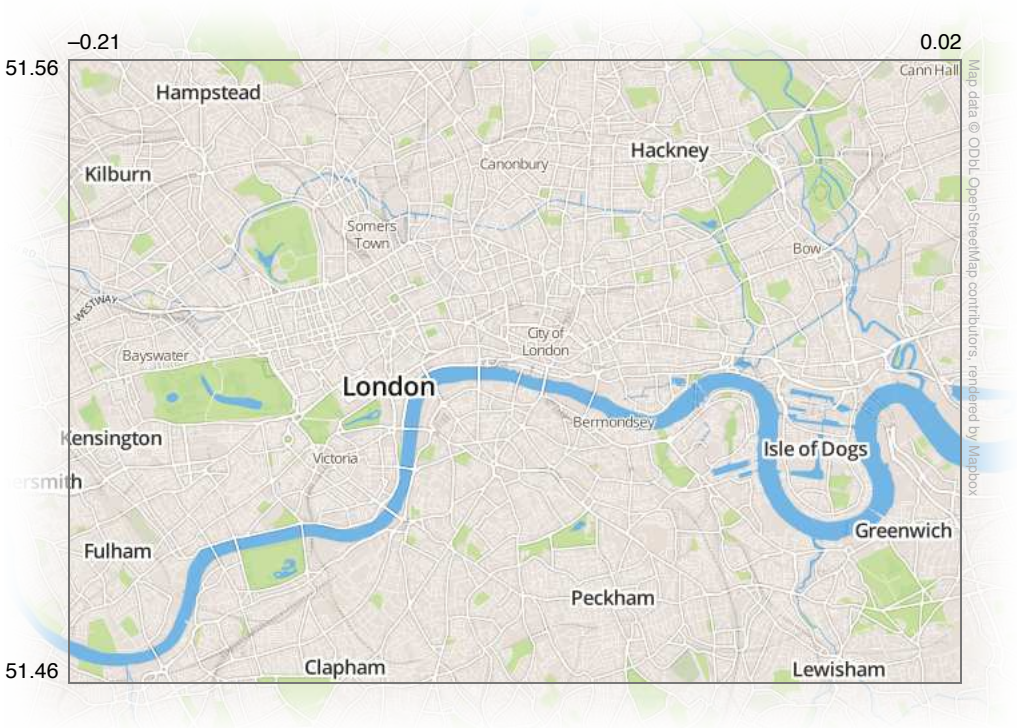


Figure 3.6: Boundaries of a region chosen for the experiments in this work.

The chosen bounding rectangle is shown in Figure 3.6. It has an area of 177.6 km^2 and dimensions of approximately 11.1 km from North to South by 15.7 km from West to East (the southern boundary is 35 meters (0.3%) longer than one in the North). The region covers most of the boroughs in Inner London and includes the majority of tourist attractions as well as Royal Parks.

3.2 Data Processing Framework

The nature of the data and steps determined in the previous section implied executing a significant number of computations during the experiments. A variety of operations were supposed to be applied on information, obtained from different origins. Photographic datasets were expected to be similar, however, it was known in advance that there would be peculiarities in their structures. These distinctions were determined by differences between their origins, some of which could not be smoothed or neglected. For example, photographs from one photo-sharing service could be supplemented with additional important information not available for images from another source. Besides, some attributes with the same semantic meanings could be encoded differently, thus requiring origin-specific methods of their processing.

The necessity to combine photographic data with the road network data in order to build a routing system was also a source for complexity. It was important to find a flexible way of supporting interrelations between datasets and avoid narrowly applicable solutions to maintain coherence.

Uncertainty in steps required to conduct the research at its beginning and a need to manipulate data from different domains suggested to look at the problem of data processing at a wider angle. It was decided to identify principles of dealing with complex interrelated data structures under conditions of action uncertainty and generalise theses principles by implementing a new data processing framework. As the research was not involving dynamic data (such as transaction flows that need to be processed in real-time), the scope of the framework was limited to static datasets (those that do not change in real time) to avoid redundant complications.

3.2.1 Problems in dealing with complex data structures

Before identifying problems that may occur during the analysis of multiple interrelated datasets, the terminology used in the discussion should be established. As the meaning of some terms slightly varies in literature, they are defined here to avoid confusion:

Collection of data is all data in the context of a given project. *Example: data behind a photo-based routing system, consisting of multiple photographic datasets and information on a street network for one or several cities.*

Dataset is a bundle of closely related data that describes some phenomenon and consists of several entities. *Examples: data behind a social network, census results, outcomes of a single experiment.*

Entity is a semantic unit of a dataset. *Examples: a user, a household, a specimen.*

Dataset component (or **component**) is a structural unit of a dataset, which can be represented in a tabular form. In most cases there is no difference between an entity and a component; however, some entities can be divided into multiple components for performance and storage optimisation. *Example: user basic info + user profile details.*

Component attribute (or **attribute**) is an atomic property of a component; it can be stored in a single column of a table. *Examples: user name, town population, voltage reading.*

Component record (or **record**) is data about one component entry; it can be represented as a single row of a table.

Dataset property is a feature of a dataset, which can be represented as a key-value pair. It does not require the creation of a dedicated entity. *Examples: social network domain name, census time period, conditions under which the experiment was held.*

Component property is a feature of a dataset component (and a corresponding entity), which can be represented as a key-value pair. It does not imply the creation of a dedicated record in a component. *Examples: id of the most recently created user, the number of census applications returned, maximum registered voltage reading.*

Primary data are components, properties and attributes that are original. They cannot be obtained from other data in the given dataset or any other dataset of the current collection of data. *Examples: user password, contents of census forms, aims of an experiment.*

Derived data are components, properties and attributes that are obtained by manipulating primary data or other derived data. All deleted derived data are restorable solely from other data in the given collection of data. Repetition of the same sequence of data processing steps always results identical derived data given the same primary data. *Examples: user password hash, median income, standard deviation of voltage readings.*

Dependency is a relationship between the derived data and the data needed to obtain them. Derived attribute *A* is dependent on attributes *B* and *C* if changes either in *B* or *C* require running certain data processing steps to update *A*. The same applies to properties, components and any combination of them. *Example: experiment summary statistics need to be updated after receiving a new portion of voltage readings.*

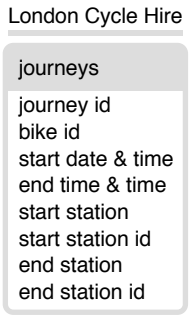
Thus, in terms given above, a collection of interrelated datasets are the datasets having derived properties, components or attributes with dependencies on data from other datasets. Combined together, they form a data collection. Data processing is a sequence of actions for producing derived data (datasets, components, component attributes, component records, dataset properties and component properties).

To understand the needs and the difficulties related to processing of arbitrary interrelated datasets, it was decided to consider several examples of collections of data from different domains. After looking at what types of tasks are performed in a wide set of cases, it was possible to summarise general data processing principles for this research and keep them coherent and clear. To discuss the obtained principles, this section looks at possible data structures and data processing tasks related to public bicycle rental schemes, such as Barclays Cycle Hire in London (`http://www.tfl.gov.uk/modes/cycling/barclays-cycle-hire`) or Velib' in Paris (`http://en.velib.paris.fr/`). The overview goes from elementary to more convoluted cases, which helps describe problems relevant to complex data structures.

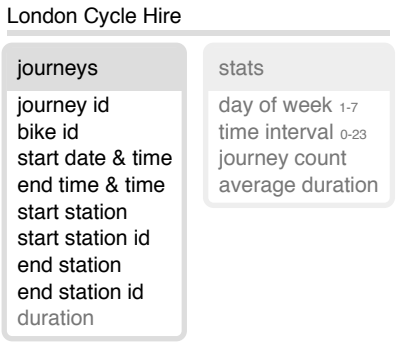
A common data processing scenario deals with a single static dataset. In the simplest case all primary data fit one component, or, in other words, a single table or one array of records. For example, these can be journeys made by users of a bicycle rental scheme in London available at `http://www.tfl.gov.uk/info-for/open-data-users/our-open-data`.
Figure 3.7a on the next page shows a a sample of raw data and Figure 3.7b describes the structure of these data in the suggested terms. Let the goal of data processing be making two histograms, one showing changes in demand over times of day and days of week and the second one with the variation of an the average journey time. This suggests: (a) adding one attribute to the primary component to store the duration of each journey and (b) creating a new component for keeping journey counts and average durations for every time bin (see Figure 3.7c).

journey id	bike id	start date & time	end date & time	start station	start station id	end station	end station id
2570	3340	30/07/2010 06:00	30/07/2010 06:22	Warwick Avenue Station, Maida Vale	47	Warwick Avenue Station, Maida Vale	47
2574	3870	30/07/2010 06:00	30/07/2010 06:14	Liverpool Road (N1 Centre), Angel	234	West Smithfield Rotunda, Farringdon	203
2578	1627	30/07/2010 06:01	30/07/2010 06:29	Kennington Road Post Office, Oval	149	Kensington Olympia Station, Olympia	293
2584	1695	30/07/2010 06:02	30/07/2010 06:06	Hampton Street, Walworth	152	Ontario Street, Elephant & Castle	324
...	...	...	...	...	...	...	...

(a) Raw primary data



(b) Primary data structure: one dataset, one component



(c) Structure for both primary and derived data: one dataset, two components

Figure 3.7: Example of a simple collection of data.

These derived data are either temporarily created in memory when data processing takes place or stored on a disk for later reuse.

If, for instance, the goal of data processing is to make a visualization of bike flows on a map, primary data is supplemented with additional information on docking stations such as geographical coordinates. The second tabular structure is used in this case to avoid redundant repetition of station-related attributes. In terms defined above, this is still one dataset as both structures are closely related and thus are the entities of the same dataset (Figure 3.8 on the following page).

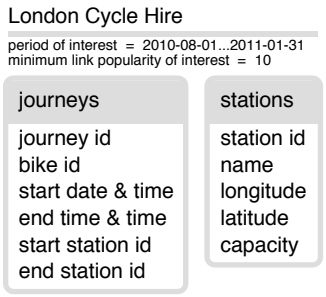
If there is no rationale for splitting any of the given entities into two or more dataset components, so the number of components is the same as the number of entities. Derived data in such a case can be colour, screen coordinates or other visual variables, needed to produce the visualization.

Working with data that forms only one dataset is not problematic, as dataset components can be easily mapped to files, database tables and variables with predefined names. These names are static and therefore can be hardcoded without any harm. The same applies to properties of the dataset and its components (if any) – these can be put into a configuration file or stored as constants and variables in code.

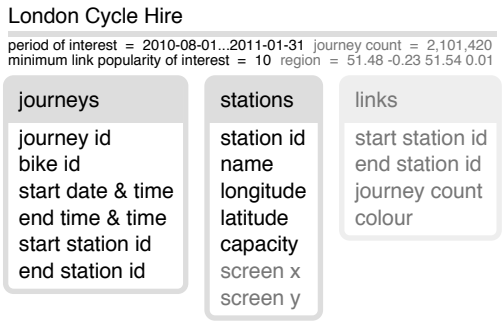
journey id	bike id	start date & time	end date & time	start station id	end station id
2570	3340	30/07/2010 06:00	30/07/2010 06:22	47	47
2574	3870	30/07/2010 06:00	30/07/2010 06:14	234	203
2578	1627	30/07/2010 06:01	30/07/2010 06:29	149	293
2584	1695	30/07/2010 06:02	30/07/2010 06:06	152	324
...	...	...	...	...	...

station id	name	longitude	latitude	capacity
73	Old Street Station, St. Luke's	−0.08848	51.52572	36
344	Goswell Road (City Uni), Finsbury	−0.10102	51.52824	17
326	Graham Street, Angel	−0.09998	51.53266	25
374	Waterloo Station 1, Waterloo	−0.11386	51.50402	32
...	...	...	...	...

(a) Raw primary data



(b) Primary data structure



(c) Structure for both primary and secondary data

Figure 3.8: Example of a collection of data, consisting of one dataset with several components.

There are cases when it is necessary to simultaneously deal with several datasets having multiple components. An example of this scenario is a bike share map showing statuses of docking stations in different cities (`http://bikes.oobrien.com/`). Here each city forms a separate dataset with its own components and properties, and the number of these datasets can be changed over time.

Having multiple datasets in one collection of data adds complexity to data processing as each tabular structure gets two coordinates: one denotes to the name of the dataset and another one identifies the component among others within the dataset. The same applies to dataset and component properties: their names or values cannot be easily hardcoded and thus need to be stored in a database or in configuration files.

Although multiple datasets contain the same kind of data, there may be differences in structures of components with the same role. For example, some component attributes can be absent in some datasets or have different formats. Similarly, some components can be missing in a number of datasets due to unavailability of particular information. Therefore, dataset properties can also consist of diverse lists of key-value pairs. Figure 3.9 on the next page

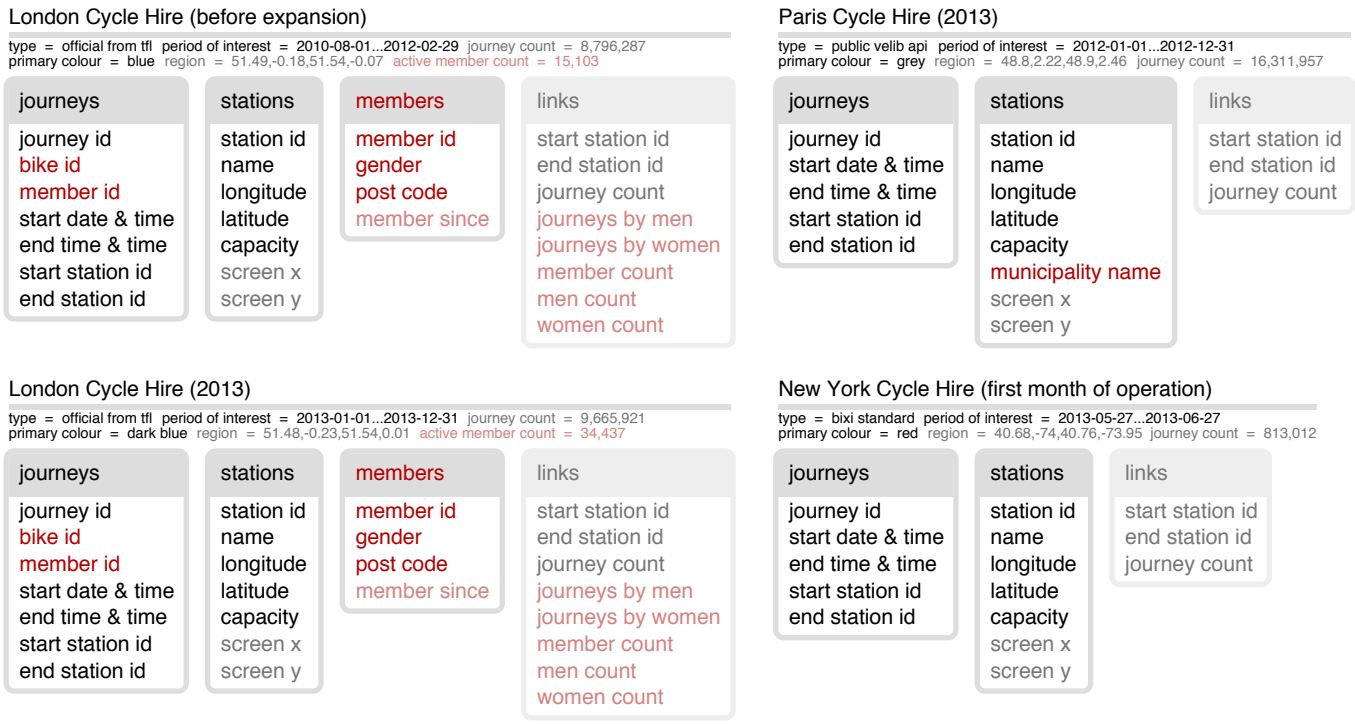


Figure 3.9: Example of a collection of data with several datasets, each containing multiple components. Dataset-specific components, properties and attributes are coloured. Shown numeric values are fictional.

helps explain these statements by showing a data structure for an imaginary project, aiming to visualize bicycle hires in multiple cities and different time periods.

When datasets have peculiarities such as distinct components, attributes or properties, their creation and processing becomes more difficult. For example, if the data are stored in a relational database, the same SQL queries cannot be applied for every dataset, because of differences in table structures and also non-static names of tables and indexes. Even if this problem does not exist due to use of other data storage strategy or is somehow resolved, all data processing routines should be aware of adaptive dataset structures and work accordingly. To the author's knowledge, there are no conventional software design patterns that are widely applied to resolve these issues.

A collection of data becomes even more complex if containing datasets are from different domains, i.e. describe entities of not the same nature. For example, to create an estimated map of cycle journeys with respect to a road network, it is necessary to supplement the data

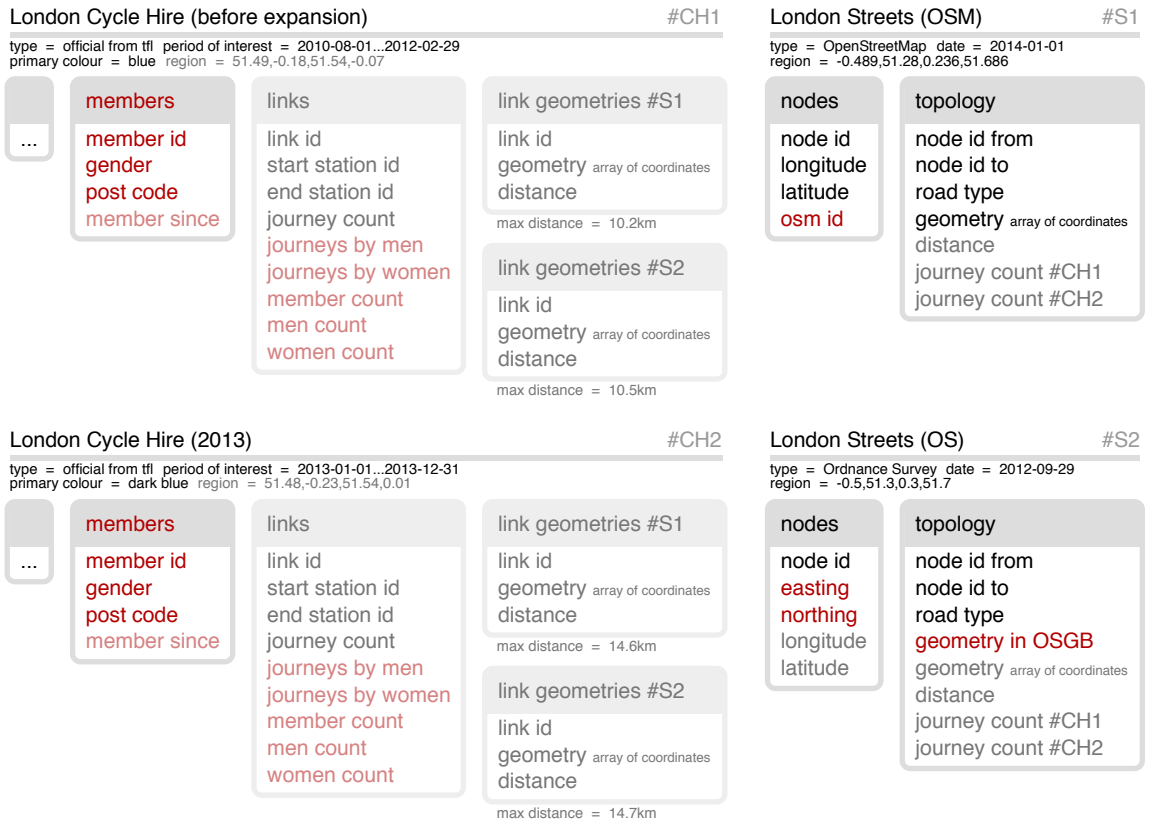


Figure 3.10: Example of a collection of data with interlinked datasets from two domains: cycle hires *(left)* and street networks *(right)*. Components and attributes with dependencies to foreign data contain hashtags as references in their names. Dataset-specific components, properties and attributes are coloured.

with street topology graphs for each city. As there may be multiple datasets on cycle journeys for the same city (e.g. London Cycle Hire 'before expansion' and 'in 2013'), it becomes rational to keep road network data as separate datasets rather than mixing them with data on bicycle journeys (see Figure 3.10).

The datasets with road networks can come from different origins and thus may also have varying properties, components or attributes, just as datasets on bicycle hires. As such datasets are interlinked, some derived data have dependencies on other datasets, and this also contribute to the complexity of a collection of data.

Now the structure of each dataset depends not only on its type, which determines some peculiarities (like in Figure 3.9 on the previous page), but also on the domain it belongs to. Besides, attributes and components can form groups: in a component there may be more than one at-

tribute with the same semantic meaning (e.g. *journey count* based on *cycle hire data one, two, three, etc.*), and there may also be multiple components storing the same sets of attributes (e.g. *link geometry* based on *street network one, two, three, etc.*). When a group of components consists of several instances, each of them may have its own properties, and these properties cannot be treated the same as those belonging to a dataset.

It is important to note that a collection of data is not necessary kept in a single place. It can be a mix of databases, files or variables, not all of which are permanent. For instance, all derived data in the second example (Figure 3.8 on page 64) can be extracted and kept only in the memory of a tool, which visualizes bicycle journeys, and thus be deleted every time the software is closed. Thus, in a general case a collection of data that belongs to a project is something having semantic boundaries rather than a single physical place, where all primary and secondary data are stored.

Uncertainty can be another source of complexity. It is not always possible to determine the structures of all datasets at the beginning of the project, because intended data processing steps may change after preliminary experiments, leading to the amendments at later stages. For instance, visualization of individual cycle journeys may appear to be slow, and this will suggest introducing a new component called 'links' with its own set of attributes (Figure 3.8 on page 64). Not all of such changes can be predicted in advance, which may require a significant amount of work on updating structures of all existing datasets and to compute new derived attributes.

The collections of data shown in Figures 3.7, 3.8 and 3.9 can be considered as special cases of what is shown in Figure 3.10. The latest example is very similar to the data behind a photo-based routing system, which can be described as follows:

> There are several datasets from two domains (photographic data and road network data).
>
> Photographic data come from different origins, which may result peculiarities in structures of datasets belonging to this domain.
>
> Datasets from both domains have spatial boundaries; photographic datasets may also have temporal boundaries, and street network data can have a version (retrieval date).

Datasets are interrelated: photographic datasets impact street network data.

Therefore, a system, which would introduce general rules for managing data structures as in the most recent example (Figure 3.10 on page 66), could be useful in this research and also find applicability in some other cases. Although numbers of projects have already dealt with the cases of similar complexity, no evidence could be found about any published solutions that would suggest a general approach for handling data with the following characteristics:

A collection of data consists of multiple datasets.

Datasets belong to a number of domains (i.e. their structures can be entirely different).

Datasets in each domain share the same structure, but can have peculiarities such as specific components, attributes or properties.

Datasets impact each other (derived data in one dataset have dependencies on data one or several other datasets).

Data structures and data processing steps are uncertain at the beginning of the project, and the plan can change over time.

According to the needs and issues described above, an 'ideal final result' (Altshuller 1999, p. 77) would be a data management system with the following features:

Any unit of data (collection of data, dataset, component, attribute, property, record) can be treated as an object and thus be be created, read, updated and deleted (Martin 1983) by executing atomic actions.

The system can be adjusted for working with a new dataset with no effort, including cases when a dataset is from a new domain or has peculiarities (i.e. distinct properties, components or component attributes).

Any changes in dataset structures can be applied immediately to all datasets, and there is no need to update any of the data processing procedures.

Any change in the software, which performs data processing, can be tested immediately without delays.

All operations on the data can be done via a single interface (a unified entry point); atomic operations can be combined into more complex routines so that it is possible to run experiments of any complexity by executing a single action when user input is not required between the data processing steps.

The system naively supports non-trivial types of attributes and properties such as geometrical and geographical shapes.

Making a mistake at any stage of data processing is not crucial; the cost of restoring any lost data is zero.

Being able to run data analysis on top of a system with these features would allow more flexibility in experiment design. For example, it would be possible to easily compute multiple instances of derived data based on various data processing rules. Treating records, attributes, components, properties, datasets or even dataset collections as objects would make any manipulations easy by avoiding a need to deal with these units on a physical level. This is especially important when uncertainties in data structures and actions take place.

3.2.2 Data processing workflow and framework concept

To understand how the processing of complex collections of data could be organised, it was necessary to find similar components of the workflow in addition to common structural units. This was needed to identify functional elements of a system that was going to be designed. Figure 3.11 on the next page shows the processes that are involved in dealing with an arbitrary collection of static data.

The contents of a collection of data (i.e. all data that can be considered as ‘internal’ for a given project) depend on three types of actions: (1) data structure determination, (2) primary data import and (3) extraction of derived data. In the simplest scenario, these processes are involved strictly one after another, and the resulting data are exported at the end. However, the order may change due to action uncertainty at the beginning of the project, inability to collect all primary

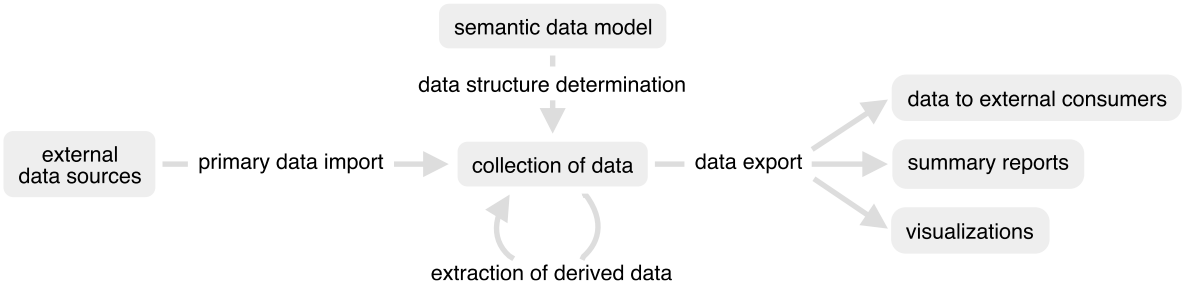


Figure 3.11: Elements of a generalised data processing workflow.

data before the extraction of derived data and for many other reasons. Use of interactive visual analytics (Wong and Thomas 2004) is a good example of what can mix the sequence of the processes involved. Doing both the extraction of derived data and their export into graphical representations, visual analytics software can also play a role of an import tool, allowing new primary data to be entered by a user.

Despite that it was impossible to unify a sequence of processes that influence a collection of arbitrary data, it was decided to extract common rules and patterns into a framework that could be reused. Such approach would imply the introduction of additional software, the purpose of which would be dealing with data units defined earlier in a number of ways regardless of the nature of the data and the order in which the actions are taken. This would not only help to reuse some parts of the code outside this thesis, but would also serve as a backbone for all manipulations of data in this research. Shared functionality that would be applicable to any dataset within one project was recognised as a means to accelerate the implementation of experiments, thus providing more freedom in their design.

"A framework is a model of a particular domain or an important aspect thereof." (Riehle 2000, p. 54). A keystone to its success is the correct identification of a domain that is being modelled. If a scope is too wide, there is less similarity between the projects the framework can be applied for, and this either increases its complexity or limits functionality. If, on the other hand, the range of cases covered by a framework is too narrow, it becomes potentially less useful as a separate unit of software.

Following the discussion in the previous subsection, the domain (the scope) of the framework was stated as 'a collection of complex interrelated datasets (excluding streaming data) that

are being processed under under conditions of action uncertainty'. As one of the main challenges it had to solve was in providing access to data with a higher level of abstraction, it was called Dataset Abstraction Framework (DAF). The development of the framework implied the following design decisions:

to establish a set of tasks the framework is responsible for;

to develop principles of interaction with data during the experiments;

to identify common data processing operations and either implement them as a part of the framework or provide a single robust interface for their implementation;

to define an approach to data storage (to understand how data units described on page 60 can be best mapped into physical data structures);

to come up with the technologies to be involved in the realization of the above ideas.

In contrast to a software library that "performs specific, well-defined operations," a framework "is a skeleton where the application defines the 'meat' of the operation by filling out the skeleton" (Cohen 2008). Thus, Dataset Abstraction Framework was to introduce a layer (Riehle 2000, p. 71) between application-specific functionality ('meat') and the rest of the software (Table 3.2):

Application Layer	functionality that is only useful in a given project
Framework Layer	functionality applicable to a range of projects that fit the framework scope
Underlying Software	platform and other functionality that is used by the above layers

Table 3.2: Layers of an application that uses a framework.

Unlike with OSI network model (ITU-T 1994), where a layer of a higher level can have access only to the closest underlying layer, software framework does not make it compulsory for the *Application Layer* to use it. The *Application Layer* can have direct access to the *Underlying software layer*, which may contain libraries or even another framework, working with a wider range of projects (Riehle 2000).

In the given context, 'meat' only includes the descriptions of dataset structures and the rules required to import, process and export the data. Actual data storage and retrieval can be

done with the help of the framework. Such segregation of responsibilities is similar to what object-relational mapping libraries (ORM) provide (Hibernate 2014; Ambler 2014). Taking project-specific descriptions of entities and relationships between them, they generate database schemas and provide access to data as linked objects. According to the author's best knowledge in October 2014, none of existing ORMs can model all data units introduced in the previous subsection; these tools are usually limited to one dataset with a predefined list of components. Besides, ORMs are not directly responsible for manipulations on the data – this is done in the *Application Layer*.

Taking into account the scope of the framework and the 'ideal final result', defined on page 68, the functionality of an arbitrary data-processing project can be split between software layers as shown in Table 3.3:

Application Layer	commons project-specific data types and models project-specific routines interface to high-level data processing routines	definition of dataset domain 1 dataset structures supported variations (dataset types) supported internal relations between data units supported external relations between datasets interface to high-level data processing routines supplements for DAF modules definition of dataset domain N
Framework Layer	DAF core data model common operations on data units interface to atomic data processing routines conventions for domain definitions	DAF module 1 data types and models routines interface to routines DAF module N
Underlying Software	programming platform data storage driver	library 1 library N

Table 3.3: Layers of an application that uses Dataset Abstraction Framework.

As all collections of data supported by the framework share the same structural units, the *Application Layer* is mainly used for their description. Here it is needed to list all dataset domains the given application can work with. The numbers of datasets in each domain and their names are not known in advance, so domain definition can only suggest a structure for any dataset that will be created later. To maintain possible peculiarities in structures within one domain, the range of the variations is also defined. A dataset can be assigned a special property *type*, which tells the framework to treat this dataset instance differently based on

a distinct definition. For example, cycle hire data for Paris may be given a type *public velib api* (see Figure 3.9 on page 65), and this will make its component *stations* contain an extra attribute called *municipality name*.

The relations between data units are also stated in the *Application Layer* in a form of executable scripts. Domain definitions contain the rules, with which to import primary data and extract derived data. This includes the description of the relations between the datasets. For example, in a case shown in Figure 3.10 on page 66, a routine that sets values to attribute *distance* of component *link geometries #S2* refers to dataset *#S2*, thus maintaining the existing interrelation.

If there are common project-specific data types, models or routines, which are not included into the definitions of the domains, they are added to the *Application Layer* as a separate group called *commons*.

Framework layer defines the general data model that is being suggested and provides access to all atomic operations on data units. It also sets the rules for domain definitions, i.e. states their format, responsibilities and limitations.

Some data processing routines may be used in more than one application, but be not common enough to become a part of the core of DAF. A report-generating tool or an approach to geographical data processing can serve as examples of such functionality. If any shared behaviour relies on the data model suggested by the framework, it is implemented as a module, which supplies additional data processing features and provides an interface to them.

Underlying Software includes a programming platform, used by the framework, and a data storage driver. Auxiliary libraries can also be added on demand. The difference between a library and a framework module in this context is in a relation to the data units. Any commonly-used piece of software is a module if it operates terms as *dataset domain, component, attribute, etc.,* and is a library if else.

In order to make DAF as close as possible to the 'ideal final result' defined on page 68, the following important data handling principles are introduced:

All project-related primary and secondary data are physically stored in one place. Although in a general case the datasets, their properties, components or even separate attributes may be distributed, DAF limits this freedom and suggests keeping all data together. This is done to maintain the coherence in the definitions of the domains and to ease access to the data. An exception can be made to some derived auxiliary variables that are only used during the export – they don't need to be placed into the storage as properties, records or attributes.

All atomic manipulations on the data are made in a form of simple operations, each affecting only one type of data units at a time. The framework receives a command and its context in a unified way (e.g. *populate component x in dataset y of domain z*), and then either executes this operation in accordance to a definition found in the *Application Layer* or by itself, if the task is context-independent. With this approach it is not necessary to create additional application-specific interfaces to process the data; any complex task can be expressed as a sequence of simple operations.

New internal data can be only obtained the following ways: setting a property, populating a component with new records or updating a component attribute. This principle applies to both primary and derived data. It makes data import and processing isolated and error-prone. Thus, there is less dependency on the dataset definition, which can be changed over time due to uncertainty. There is always only one way of obtaining any piece of primary or secondary data.

Repetition of any operation results the same new data given identical initial data. Accidental errors in a routine that imports new primary data or computes derived data do not harm the dataset. The same operation may be repeated until its execution is successful.

Dataset components are created on demand right before they are needed. The framework tries to minimise the cost of changes in domain definitions and data processing steps, which are uncertain. New datasets are created empty, and their components are added on the fly right before the corresponding data are ready to populate them. If

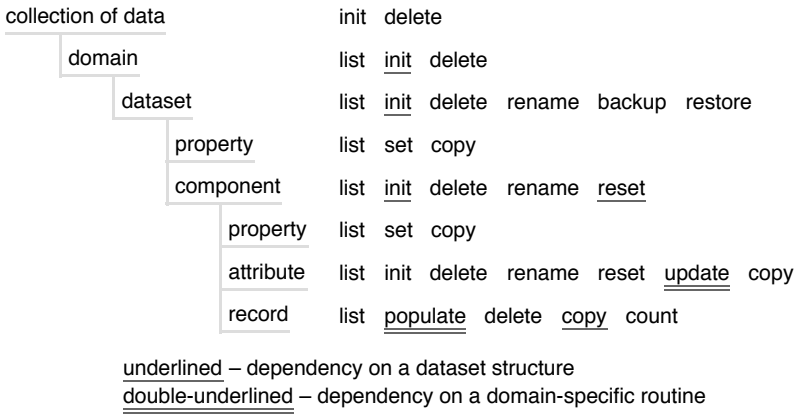


Figure 3.12: Full map of data units supported by Dataset Abstraction Framework and operations applicable to them.

a definition of a dataset domain changes after the creation of a component, its structure is either updated by adding or removing attributes or by migrating all existing data into new components. Such approach also helps saving disk space.

One application instance can only work with a single collection of data at a time. A collection of data is at the top level of the data unit hierarchy; it can fit any number of datasets and domains. Hence, there are no particular benefits of working with several collections of data simultaneously. Having only a single instance of this data unit in any project helps avoid redundant complications and confusions. Switching between collections of data is subject to application configuration.

The list of all atomic operations DAF can perform is shown in Figure 3.12.

Most of the operations are independent from the application context and therefore can be implemented on the *Framework Layer* in full. Examples of such operations are: *resetting an attribute* (setting values in all records to `NULL`), *renaming a dataset* or *copying component properties to another component.*

Seven out of thirty-four atomic operations rely on the knowledge about the data the given application is working with. Five of them require the descriptions of the dataset structures. For example, to initialise component x in dataset y belonging to domain z, the framework

needs to consider two things: a definition of a standard dataset in domain z and structural peculiarities for type x in domain z.

Only two operations (*update an attribute* and *populate records*) require domain-specific routines. These pieces of software, which belong to the *Application Layer*, are executed by the framework in respect to the given context. Both can involve complex algorithms that either import primary data or extract derived data.

The sequences of actions can be combined together. These high-level application-specific routines are accessed similarly to the atomic operations. An example of such a routine can be the following sequence: *create a new dataset with these parameters, collect primary data using some API, calculate statistics, import more data based on the derived statistics, calculate more statistics, generate three reports*. Thus, if processing of one dataset can be done with no interruptions for human input, the operations can be merged and executed by sending only one command to the application. If multiple datasets need to be handled the same way, it is possible to alter one parameter in a command and submit it again.

3.2.3 Technical implementation

Minimisation of the distance between the 'ideal final result' on page 68 and the suggested solution is achieved not only with a carefully defined scope of the framework and the principles it follows, but also on a chosen set of technologies that need to be involved.

After the concept of Dataset Abstraction Framework was well-established, it was necessary to select the programming platform and the approach to data storage. These two important constituents are the crucial, as they determine the flexibility and performance of applications built on top of the framework, including the one needed for this research.

The task of the **data storage engine** is to organise the project's internal data (i.e. a single collection of data). The smaller the difference between the data units defined by the framework and those that are native to a particular storage type, the easier it is to map them in both directions. As the suggested data model consists of tabular structures, the choice of the stor-

	SQLite 3.8	MySQL 5.6	PostgreSQL 9.3
data units	file (database) table / view data column data record	schema (database) table / view data column data record	database schema (namespace) table / view data column data record
support of custom data types	no	yes	yes
support of geographical data types and indexes	no	partial (OpenGIS)	advanced (PostGIS)
serverless	yes	no	no

Table 3.4: Comparison of candidate data storage engines for DAF. Sources: (SQLite 2014; Oracle 2014; PostgreSQL 2014).

age engine was narrowed down to relational database management systems. The comparison of the most popular open-source solutions (DB-Engines 2014) is shown in Table 3.4.

It was decided use PostgreSQL (`http://postgresql.org/`) for a higher number of data units this engine can operate and for the advanced support of geographical attribute types provided by PostGIS (`http://postgis.net/`), one of its extensions.

Framework data units are mapped to PostgreSQL data units as shown in Table 3.5. Every collection of data is a separate database, which can be accessed only by the specified owner or the server's root user. As a single instance of PostgreSQL can contain multiple databases, more than one DAF application can safely host data on it. Domains correspond to schemas (namespaces), which are local clusters of tables, views, functions and types. The problem of mapping two remaining coordinates of tabular structures in DAF (dataset name + com-

DAF	PostgreSQL
collection of data	database
domain	schema
dataset	table `[dataset_name]__meta`
component	table `[dataset_name]__[component_name]`
component record	table row
component attribute	table column
dataset or component property	key / value pair in `[dataset_name]__meta`

Table 3.5: Mapping between DAF and PostgreSQL data units.

ponent name) into only one coordinate left in PostgreSQL (table name) has been resolved with a naming convention for tables. Dataset name is always followed by two underscores, after which there is a name of a component. In order to avoid collisions in this mechanism, the framework checks all names against containing this sequence of characters. Dataset and component properties are stored as key-value pairs in a table called `[dataset_name]__meta`. The keys for component properties are also kept in this table; they start with the name of the corresponding component, followed by a period. When table `[dataset_name]__meta` is empty, it works as an indication that a dataset with a corresponding name exists. Component attributes and records are table columns and rows, respectively. The cases of having groups of attributes are also handled by adding two underscores in the middle of the column name. The same applies to groups of components in a dataset – a table can be named as `[dataset_name]__[component_group_name]__[component_instance_name]` instead of `[dataset_name]__[component_name]`.

An example data unit mapping is shown in Figure 3.13.

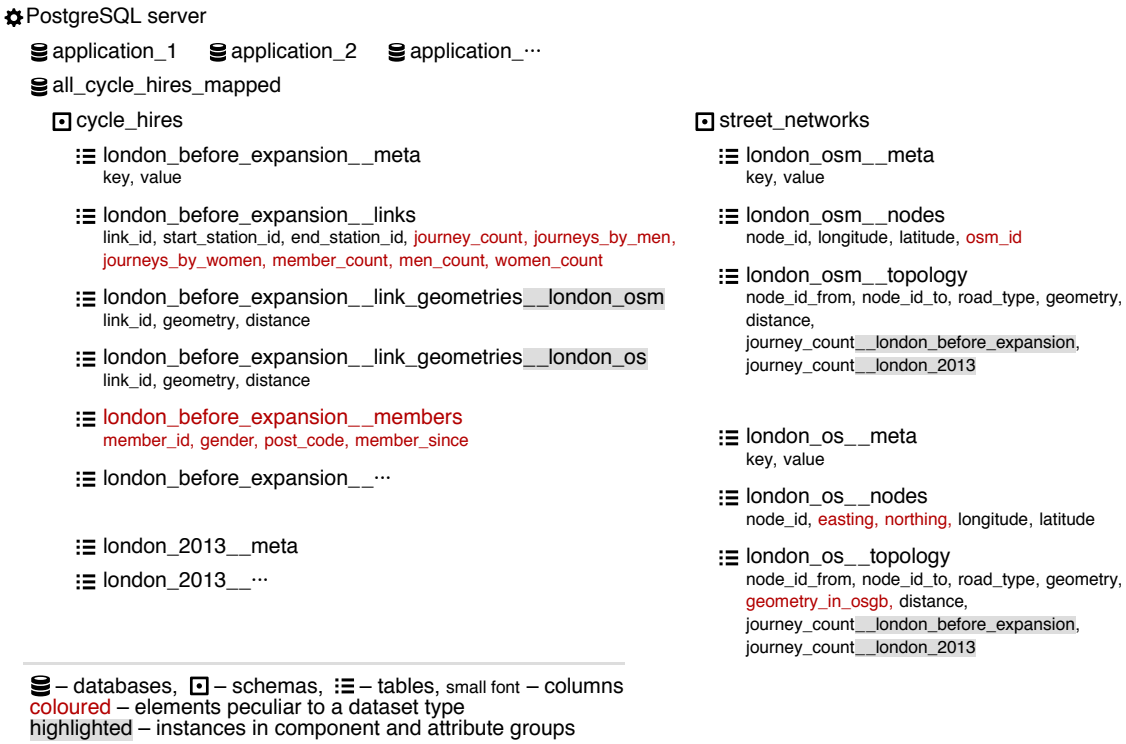


Figure 3.13: A collection of data from Figure 3.10 on page 66 mapped into PostgreSQL units.

The simplicity of the chosen approach makes it easy to access a DAF-based collection of data with the tools that do not support all concepts, suggested by the framework. Thus, a number of specific tasks in a data processing project can be delegated to some auxiliary software that is not familiar with DAF.

The only important limitation, which is brought by the proposed concept-to-technology mapping strategy, is related to the names of the data units. PostgreSQL does not allow table and column identifiers to exceed 56 symbols (PostgreSQL 2014), and this may become problematic when datasets contain groups of components or attributes. It is the developer's responsibility to make sure that the names of certain data units are not long enough to face this limitation. The set of allowed characters consists of small letters, numbers and underscores, but the underscores can not be used one after another.

The **programming platform** is defining the environment in which two other software layers are operating. The more functionality is delegated to this layer, the easier it becomes to implement the framework and the framework-based applications. As DAF is designed to be used under the conditions of uncertainty, it was important to choose a platform considering not only its potential performance, but also the flexibility and dynamism it provides.

High-level programming languages are traditionally divided into two groups depending on how the software is executed (Scott 2009, p. 16). *Compiled languages* are first translated into the machine code that is then ran natively, while *interpreted languages* always require an interpreter that converts abstract statements into the machine code on the fly. *Compiled languages* are generally characterised with higher performance (Fulgham and Gouy 2014), but the cost of this benefit is time needed for a translation to complete. In a general case the duration of this process is proportional to the overall size of the program rather than to the significance of a change. When some software is used under the conditions of uncertainty and is therefore being constantly changed, the gain of higher performance can be decreased or even reduced to zero by the waiting time needed to convert new changes into the machine code. For this reason the preference was given to an *interpreted language*.

It was decided to choose PHP 5.4 (`http://php.net/`) as a core of the programming platform, and to supplement its functionality with Symfony 2 (`http://symfony.com/`), a set

of reusable PHP components that are most commonly used as a framework for web applications. The following core features of Symfony 2 significantly reduce the effort, needed to implement DAF and DAF-based applications:

File and code structuring conventions. All parts of an application built with Symfony 2 are grouped into what is called *bundles*. A bundle is a unit of software that has a clearly defined purpose, a common internal structure and a standard interface. Bundles can have dependencies on other bundles by relying on their resources.

Command line engine. A Symfony-driven web application may be maintained via the console, and this method of interaction can be used on its own as the only entry point to the functionality of the software. The engine provides means to easily add new commands and to combine them into sequences.

Service container. To enable access to the features implemented by different bundles, Symfony 2 adheres *Service-oriented architecture design pattern* (The Open Group 2010). Any object can be registered as a *service*, and then used in any part of an application that has access to the service container.

Configuration engine. Symfony suggests keeping all parameters and constants separately from the code in human friendly *yaml files* (`http://yaml.org/`), making it easy to configure applications and bundles.

Templating engine. To help with the generation of complex HTML pages, Symfony uses *Twig* (`http://twig.sensiolabs.org/`), a powerful templating library. Twig can be also utilised for rendering of all types of other text templates, including those that contain definitions of DAF data units.

Native support of PostgreSQL. Symfony 2 is shipped with *Doctrine* (`http://www.doctrine-project.org/`), a group of libraries that are focused on database storage and object mapping. Configuring one or several databases is done by adding their parameters to a configuration file.

Table 3.6 on the facing page shows the structure of a DAF-based application in Symfony 2 terms.

Application Layer	ApplicationCommonsBundle project-specific classes project-specific services project-specific commands	Domain1Bundle templates (Twig/PostgreSQL) services that update attributes services that populate components domain-specific commands templates and services for DAF modules DomainNBundle
Framework Layer	DatasetAbstractionBundle (DAF core) templates (Twig/PostgreSQL) services and classes that manipulate data units core framework commands conventions for application-specific bundles	Module1Bundle templates classes and services commands ModuleNBundle
Underlying Software	Symfony2 core command line engine service container templating engine configuratio engine PostgreSQL server	Library1Bundle LibraryNBundle

Table 3.6: The structure of a DAF-based application in Symfony 2 terms.

The core of DAF is implemented as a single Symfony 2 bundle. It describes the suggested data model in general terms, contains interpreting tools for domain definitions and also provides access to atomic operations from the list in Figure 3.12 on page 75. The interaction with a user is performed via a standard Unix, Linux or Windows terminal by means of a Symfony 2 console. For example, this is how one can create a dataset with the London's street network based on OpenStreetMap data, assuming that the definition of the corresponding domain is correct:

```
$ cd application-directory
$ app/console daf:datasets:init street_networks.london_osm osm
Initialising dataset street_networks.london_osm... Done.
Setting property type to osm for street_networks.london_osm... Done.
$ app/console daf:datasets:components:properties:set street_networks.london_osm
    bounds_spatial "bbox(-0.489,51.28,0.236,51.686)"
Setting property bounds_spatial to bbox(-0.489,51.28,0.236,51.686) for
    street_networks.london_osm... Done.
$ app/console daf:datasets:components:init street_networks.london_osm topology
Initialising component topology in dataset street_networks.london_osm... Done.
$ app/console daf:datasets:components:records:populate street_networks.london_osm
    topology
Populating component topology in dataset street_networks.london_osm...
(details of steps involved in the process are omitted)
Done.
```

A more detailed description of the commands the framework provides can be found in Appendix B on page 290.

The Application layer consists of one or more bundles, each describing a single dataset domain. When there are several domains with shared functionality, an additional linking bundle can be introduced. The framework distinguishes bundles containing domain definitions and automatically searches for corresponding routines or templates when necessary.

With Twig, the structures of tables that map dataset components are described in a rather short form. The templates inherit a base template and only contain the names of PostgreSQL columns, indexes and comments that need to be created. The columns can be enumerated using loops, which makes it possible to define groups of attributes with no need to mention all of them:

```
{# DemoDomainBundle/Resources/views/pgsql/demo_component/init.pgsql.twig #}
{% extends 'KachkaevDatasetAbstractionBundle:pgsql/bases:createTable.pgsql.twig' %}

{% block tableName %}{{ datasetName }}__demo_component{% endblock %}

{% block fields %}

    id int NOT NULL,

    {% block type_specific_attributes %}
    {% endblock %}

    {% set days = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun'] %}
    {% set months = ['jan', 'feb', 'mar', 'may', 'jun', 'jul', 'aug', 'sep', 'oct',
        'nov', 'dec'] %}

    {# some_count__mon_jan, some_count__mon_feb, ... some_count__tue_jan, ... #}
    {% for day in days %}
        {% for month in month %}
            some_count__{{ day }}_{{ month }} real DEFAULT NULL,
        {% endfor %}
    {% endfor %}

    some_attribute_1 character varying,
    some_attribute_2 character varying

{% block constraints %},  CONSTRAINT {{ block('tableName') }}_pkey PRIMARY KEY (id){%
    endblock %}
```

The attributes that are peculiar only to one type of datasets are defined in a separate template, which is automatically invoked by the framework when applicable.

```
{# DemoDomainBundle/Resources/views/pgsql/demo_component/init.demo_type.pgsql.twig #}
{% extends 'DemoDomainBundle:pgsql/demo_component/init.pgsql.twig' %}

{% block type_specific_attributes %}
    some_attribute_1_peculiar_to_demo_type real,
    some_attribute_2_peculiar_to_demo_type real
{% endblock %}
```

Domain-specific routines are registered as services with specific tags. When DAF receives a command to populate a dataset component or to update an attribute, it finds a corresponding service for the given context and invokes it. The developers do not have to write all routines in PHP – it is possible to delegate the procedures to external scripts when they are significantly less time-consuming. For example, this approach can be applied when crawling data using multiple threads, which are not supported by PHP in full (Watkins 2014).

Some functionality can be too specific to be included into the core of the Dataset Abstraction Framework, but can be still reused in a number of projects. In such cases it is recommended to implement the features as general-purpose DAF modules, which have dependencies on the framework, but do not have to become parts of all applications using it. The modules that have been created for this research are briefly described below.

3.2.4 Modules

Geographic data processing

The analysis of crowd-sourced photographs and street networks involve data processing algorithms that deal with spatial relations. As the performance of these algorithms depends on the volume of the data, working with large datasets at once may become resource intensive and time-consuming. This problem is commonly solved by splitting the area of interest into regions and processing them one after another or in parallel (Hawick, Coddington and James 2003; Migliorini et al. 2011; Aji et al. 2013). Existing methods differ by the space decomposition strategy, the order of the operations and the overall applicability. In cases when the spatial distribution of data is not even, it is more efficient to work with clusters of varying size than with grid cells having equal dimensions. This approach allows a data processing algorithm to allocate sufficient resources to dense areas while going through less populated ones at a higher speed.

The method of space decomposition chosen for this research is inspired by Tchaikin (2008). Attempting to collect the locations of photographs from Panoramio, he observed that the API was not handling the requests for large or dense areas in a proper way. This resulted in absence of some data records in the responses or even failures in retrieving them. He suggested to use

a dynamic top-down approach, wrapping an arbitrary area of interest into a bounding box and then recursively splitting this rectangle into four sectors in order to make each of them small enough to be handled. First, the sectors were decreased in size until the API could report the quantity of photographs they contained. If this number was bigger than a threshold, the process continued until the threshold was reached or a region became indivisible. Thus, it was possible to both maximise the number of reported photographs and minimise the load on the API, saving time as a consequence.

The screenshot of Google Earth (`http://earth.google.co.uk/`) in Figure 3.14 shows how this approach was applied to collect Panoramio images in Europe. As it is seen from the visualization of the sectors, unpopular areas such as those covered with water were handled by means of only a few API requests, while large cities and other dense areas were given more attention.

Data structures in which each internal node has exactly four children are called *quadtrees* (Finkel and Bentley 1974). They are used in a wide range of tasks not limited to those that are geography-related (Samet 1990*a*,*b*). Taking into account the previous successful experience, quads were chosen to serve as a backbone for all routines, involving spatial data processing in this research. A generalised method of sector division and handling was implemented as a Dataset Abstraction Framework module.

Figure 3.14: Europe, divided into sectors after using a dynamic top-down approach for crawling metadata from Panoramio. *Source: Tchaikin (2008).*

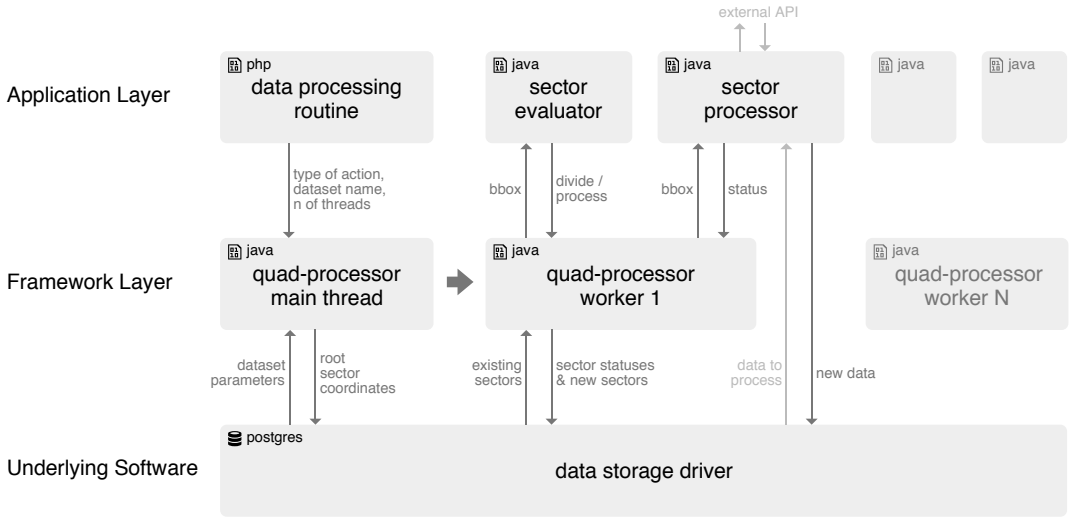


Figure 3.15: Quad-processing DAF module. Optional connections are faded.

The quads are independent from each other, which enables their handling in parallel. If there are any objects that cross the boundaries of the sectors, they are processed more than once as suggested by Aji et al. (2013, Figure 5). Because threading in PHP is only supported by an experimental extension (Watkins 2014), the implementation of the tool has been mostly done using Java (`http://www.oracle.com/technetwork/java/`). The internal structure of the *quad-processing* module and the workflow it suggests are shown in Figure 3.15.

An application-specific routine (which either populates a dataset component with records or updates an attribute) launches an instance of *quad-processor* and passes the configuration to its main thread. The tool initialises a new component, adds a root sector (if needed) and then launches a pool of workers, the task of which is to divide and process all unconsidered sectors. All boundaries are defined in EPSG:4326 (i.e. WGS-84). A free worker reserves a sector and sends its coordinates to the *Application layer*, where it is decided whether or not the sector needs to be split further. When the decision is finally made to process a sector, the worker changes its status and waits for the operation to complete. Lastly, the worker closes the sector and proceeds to the next available one. Any unexpected error marks a sector as faulty and terminates its processing.

After all sectors are considered, an operator of a data processing application checks for faulty sectors and either resets them to run the same routine again or attempts to fix an issue if it is caused by the errors in the software.

Figure 3.16: Types of sector division and subsector naming conventions used by the *quad-processing* DAF module.

When the dimensions of a root sector are arbitrary, splitting it into exactly four parts can be inefficient in rare cases, in particular when the proportion between the width and the height is extreme. This potential issue is resolved in the module by adding two supplementary types of sector division as shown in Figure 3.16.

The chosen approach for spatial data analysis fits all relevant tasks in this research and can be also applied in a wider variety of cases.

Data processing using R

Despite that a number of statistical methods are natively supported by PHP 5 (`http://www.php.net/manual/en/ref.stats.php`), this environment is not the best choice for performing complex analyses. The reasons is a relatively low performance and lack of some important and widely applicable algorithms.

Use of specialized statistical packages outside DAF is also problematic. This violates the idea of a single entry point to all data processing tasks, suggested by the framework. As a consequence, it becomes impossible to run a sequence of required routines by calling a single command in some situations. Besides, because a collection of data consists of several datasets with arbitrary names, applying the same analyses to all of them requires manual changes in parameters of statistical functions. This is not error-prone and may be time-consuming.

A solution was found in linking Dataset Abstraction Framework with R, a free programming language and software environment for statistical computing (`http://www.r-project.org/`).

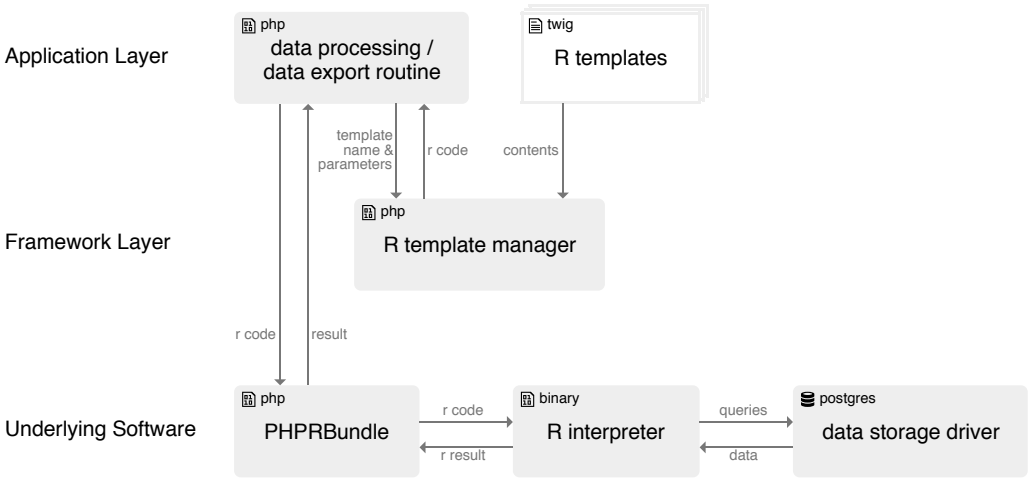


Figure 3.17: Integration of R into DAF.

The integration is done by means of a DAF module and a library, which work as a bridge between PHP and R interpreter (Figure 3.17). R code is stored in a form of Twig templates inside the Application layer. A user-defined routine renders the required templates with *R templating manager* and passes the resulting R code to PHPR (`http://github.com/kachkaev/php-r`):

```
{# DemoDomainBundle/Resources/views/r/demo_r_code.pgsql.twig #}
...
records <- dbGetQuery(con, statement = paste(
     "SELECT a, b, c, d",
     "FROM demo_domain.{{ datasetName }}__demo_component",
     "ORDER BY a"));
...
```

↓

```
...
records <- dbGetQuery(con, statement = paste(
     "SELECT a, b, c, d",
     "FROM demo_domain.demo_dataset__demo_component",
     "ORDER BY a"));
...
```

Compiled R commands are sent to the interpreter, which queries the database, computes the required statistical measures and returns the result. After the routine has received the values of the variables it needed, it either uses them to update some derived data or to prepare an export.

With such approach it is possible to apply the same analyses to several datasets at no additional cost. An update in R code can be instantly tested on any data, which is important under conditions of uncertainty.

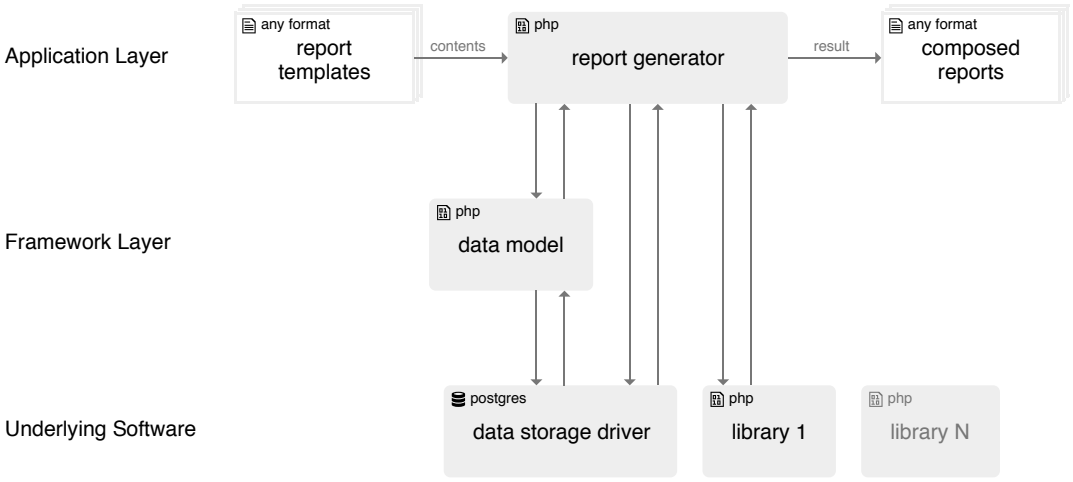


Figure 3.18: Report-generating DAF module.

Summary report generation

The results of data processing are often presented in a form of human-readable reports with text, tables, charts and other common means of information representation. These reports are either created by hand or with the use of specialised report generator software.

When a collection of data consists of multiple datasets, it is rather probable that the same report type is applicable to many if not to all of them, including those that might be created in future. Errors or uncertainty in the data processing steps, which go prior to report composition, may also take place. These conditions suggest that in DAF-based applications it is more efficient to generate reports rather than to compose them manually, and for this purpose another framework module has been introduced.

The module is not aiming to propose any standards for the contents of the reports – it only establishes a common workflow to the procedure of their creation (Figure 3.18).

The bundles in the Application layer register special services, each knowing how to generate a report with a certain name. When a user runs `reporting:generate` and passes the name of a report, the module automatically searches for a corresponding service and launches it if this service exists. The result is saved in a pre-configured output directory. The names of the files are versionied (i.e. *report-name_DATE-TIME.format*), so previously generated reports are not deleted unless requested. Report names can be suffixed with modifiers, which may (but don't have to) correspond to dataset names. For instance, a file to generate can be called

street-network-summary__london_osm.txt or *routing-graph-comparison__all.xls* (the first example is a text file with a street network summary for dataset *london_osm*, and the second one is a spreadsheet with a comparison between all existing datasets that contain routing graphs). The module does not define how the suffixes in the report names are handled – it is the responsibility of the services to 'understand' their meaning.

The applications are free to choose between any appropriate report formats, such as plain text files, Excel sheets, Word documents, HTML pages, etc. The list of all available report names, modifiers and formats can be shown to a user by calling `reporting:list`.

A more detailed description of DAF modules can be found in Appendix A.

3.2.5 Usage example

The core of Dataset Abstraction Framework was successfully used in a web-based interactive data visualization in 2013. The software was a part of a project conducted at giCentre for E.ON (Goodwin et al. 2013). The purpose of the application was to explore household electricity consumption in a form of 'energy signatures' (Figure 3.19 on the following page).

Primary data were the output of a modelling tool (Gruber and Prodanovic 2012) that generated CSV files with electricity readings for individual appliances in a set of imaginary households. These fictional readings were covering a certain period of time at equal 15-minute intervals. The model was run more than once with different settings, which produced multiple datasets that needed to be processed and visualized; the size of the largest one was 2.8 GB. The challenge was to aggregate the readings by appliance type, time of day and day of week to show variations in energy consumption across these groupings and also to facilitate comparisons between datasets.

The structure of the derived data was rather complex: there were five statistical measures (minimum, maximum, mean, median and standard deviation) for each of 24 variables (Table 3.7 on the next page) related to 85 individual appliance types and 57 groups of them; datasets had their own modelled readings. Statistics were needed to be calculated not only for all datasets,

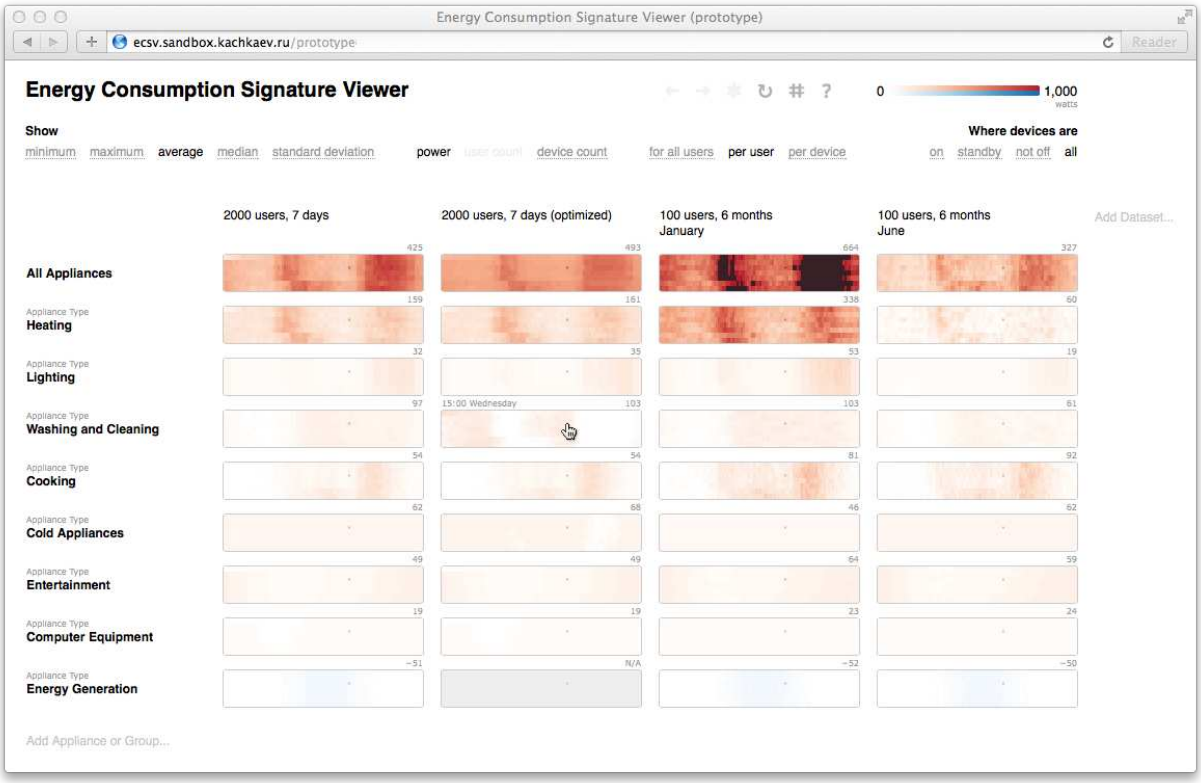


Figure 3.19: Energy Consumption Signature Viewer, a web application that uses DAF. Available at `http://ecsv.sandbox.kachkaev.ru/`.

but also for any chosen subset (e.g. only houses with electric heating, second week of May). The results of data processing had to be accessible instantly on user's request.

Using framework's terms, the given collection of data was consisting of one domain of datasets all having the same type (there was no variation in names of attributes or unique components, specific only to some datasets). All datasets were independent. Each of them contained two

	power	devicecount	usercount
perdevice	all standby on not0		
peruser	all standby on not0	all standby on not0	
perweek	all standby on not0	all standby on not0	all standby on not0

green – applicable to individual appliances only (null for groups)
gray – not precalculated (extracted from primary data on fly)

all – all devices, standby – only devices on standby (TV sets etc.), on – only devices that are on, not0 – on + standby

examples: power_perdevice_all, usercount_perweek_standby

Table 3.7: A grid of variables for which five statistical measures were calculated in Energy Consumption Signature Viewer.

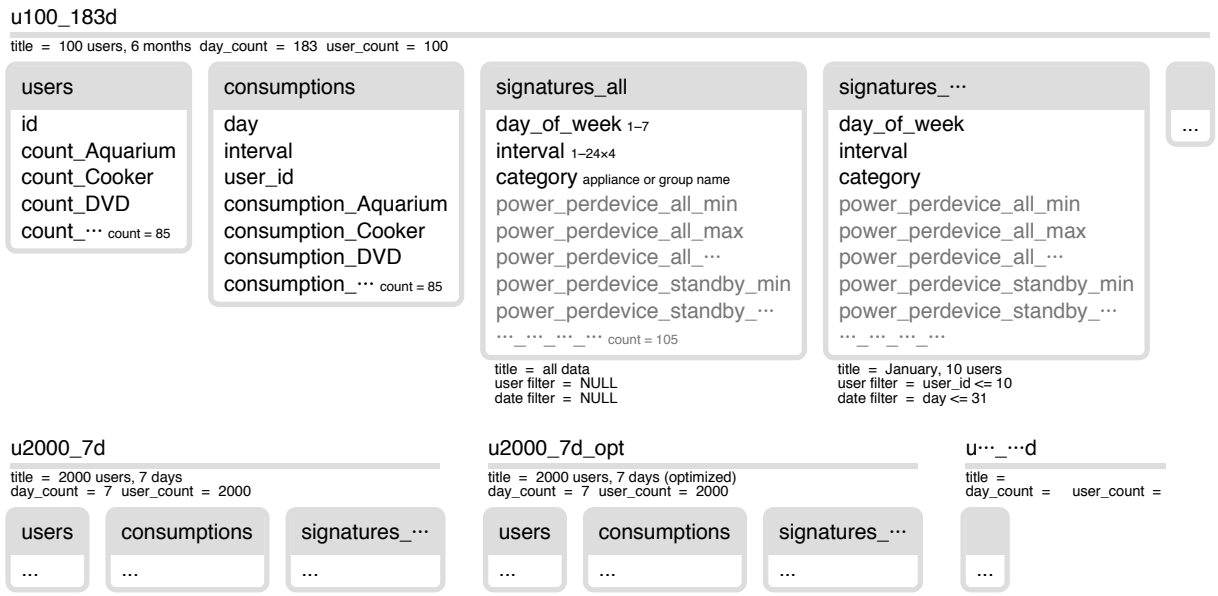


Figure 3.20: The structure of data in Energy Consumption Signature Viewer, an application that uses Dataset Abstraction Framework.

components with primary data and one component group with derived statistics (Figure 3.20). As these statistics were time-consuming to obtain, they were calculated in advance on the server side of the application. When a user switched to a new measure or added either an appliance or a dataset, the client requested values of a particular attribute and for a certain group of records in all corresponding components. Despite that each component instance contained $6,701,310$ derived measures (105 attributes and $63,822$ records), the data could be instantly retrieved, delivered to the client and shown in a browser.

DAF made it easy to maintain tens of tables with over a hundred of columns in each, which could become difficult and confusing without its use. PostgreSQL queries (table creation, data insertion and retrieval) were generated on fly using Twig templates, which helped separate application logic and physical database structure. All data processing steps could be easily repeated with no changes in code for any new set of CSV files, and this ability was used multiple times when the modelling tool was adjusted by the provider. A single command-line entry point to the backend of the application significantly reduced the amount of effort and time spent during the experiments and also helped with setting up the production server. Having datasets, dataset components, their properties and attributes available in the code as objects made the process of software implementation quicker too.

The application was highly acknowledged by both the customer and the vendor of the data. Given that the project was only about a month long, Dataset Abstraction Framework was a keystone to its success. This case confirmed the wide applicability of the framework and also helped its improvement, which had a positive impact on data processing workflow in this PhD research.

The source code of the Dataset Abstraction Framework is available on github as a Symfony 2 bundle: `https://github.com/kachkaev/KachkaevDatasetAbstractionBundle`.

3.3 A visual analytic approach to data analysis

Summarising the content of the previous sections, the data in this research can be described as following:

> There are two collections of data: the first one containing photographic datasets and road networks and the second one consisting of responses from survey participants and some attributes of a sample of images (survey subjects).
>
> Size, complexity and uncertainty in the first collection of data are higher, and therefore it should be handled with the Dataset Abstraction Framework. The second collection of data consists of only one dataset, a structure of which can be determined in advance, so the use of DAF is not necessary.
>
> The entities in the first collection of data have spatial attributes: coordinate pairs for photographs (points) and edge geometries for roads (polygonal chains). The second collection of data has little geographic information (locations of photographs are also known, but they are insignificant).
>
> The operations on both collections of data are not known beforehand. They depend on the contained patterns, many of which remain undiscovered at the beginning of the experiments.

These characteristics make it difficult to apply computational methods and achieve the goals of the research without a human intervention to the process of data analysis. The development

of models, which help find patterns, as well as the assessment of the results of experiments become challenging, as these procedures cannot be easily automated and require more data models and processing themselves.

A visual analytic (VA) approach is what is often applied in similar cases. This is a multidisciplinary method that combines computational techniques and human judgement to "detect the expected and discover the unexpected from massive and dynamic information streams and databases" (Thomas and Cook 2006, p. 10). By combining visual representations of data and interaction, visual analytics software tools delegate pattern detection to a human brain, thus eliminating a needed to develop complex mathematical models that would be otherwise needed to describe the subject under study. Although this approach cannot substitute statistical methods as it does not produce measurable results, it significantly reduces the amount of effort required for complex data analyses.

In a visual analytics process, which is shown in Figure 3.21, the data, the visualizations, the models and the obtained knowledge are tightly linked. The approach does not suggest a framework for actions in a research – the order of steps at each stage is determined by the current goals and the results of previously conducted experiments. Visualizations can be used to propose and test new data models the same as existing data models can suggest new visual mappings. The knowledge gained during one iteration can be used to make changes in a source collection of data, and the process can be repeated again.

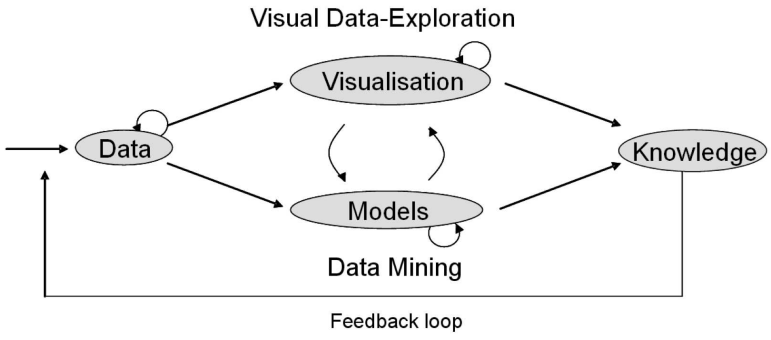


Figure 3.21: The visual analytic process. *Source: Keim et al. (2008, p. 156)*

A visual analytic approach to data analysis was found beneficial at each stage of this project:

Photographic data gathering. With VA it is possible to do a quick assessment of the collected data and detect API problems, just by looking at spatial distributions of images.

Development and testing of image filtering methods (i.e. bias-reduction functions). VA allows the anomalies in spatial distributions of photographs to be detected with ease – no development of complex mathematical models is necessary. It is also possible to analyse the effect of any suggested filtering method using simple visual comparisons of the cleaned datasets with their original versions.

Survey data analysis. Subjective responses from users may contain hidden patterns and anomalies, and their exploration with visualization can support knowledge extraction.

Road network data collection and pre-processing. Although this research was not aiming to perform the analysis of the road network data, basic integrity check would still have to be made. The ability to see all edges and nodes in a routing graph can replace complex algorithms, which would be otherwise necessary to assess these data.

Assignment of attractiveness scores to road network edges. The impact of changing the way the attractiveness scores are assigned to road networks is hard to be analysed automatically. By adding visualization into the experiments, it is possible to make judgements quicker and be more confident in their reliability.

Experiments with routing. The structure and the quality of the obtained routes can be only assessed visually, as crowd-sourced 'field studies' are not planned in this research. Besides, VA can also help monitor the workflow in the routing algorithm and ensure its correctness.

As the development of a data visualization platform, which could be used throughout the research, would require a significant amount of resources, it was decided to use existing solutions where possible and implement narrowly applicable software when no alternatives could be found.

It was important to make sure that human-facing interfaces of the new applications serve their purpose, i.e. maximise the efficiency of visualization as a tool. At giCentre, we tend to adhere

a number of well-established general design principles while developing the interfaces for visual analytic software; these principles were widely used in this research project too:

The data are the interface. We reinforce the link between data and interaction by making the elements that represent data also serve as the user interface. This reduces the need for separate buttons and menus which may cause clutter and decrease the 'data:ink' ratio (Tufte 1983). The amount of noise must be brought to minimum to be able to accommodate large volumes of information in a single view (Crampton 2002).

Consistency of encoding. We map visual encodings to data consistently, e.g. using a particular visual variable to represent a particular parameter of feature across all linked views. The same rule applies to interaction.

Use of states. Any change in the data view is considered as an action and can be reverted or repeated. Such an approach helps us to navigate quickly between states, which eases the accomplishment of certain tasks. With the use of this approach the apps can also restore the history of states after reloading, which is helpful while debugging and collaborating (Walker et al. 2013).

High information efficiency of user actions. We try to avoid user input that carries little or no information (Raskin 2000). For example, we don't use dialog boxes with only one possible action (e.g. 'OK' button).

Use of keyboard shortcuts. We find this method of interaction more appropriate in expert-oriented visual analytics applications compared to mouse interaction with interface elements and therefore use it for a wide range of actions (Fry 2008). In addition to improving the speed of interaction, we also escape from a need of having non-informative interface elements, which supports the first design principle.

Transitions between views. We use smooth animated transitions between views where applicable in order to better see the differences in the data or to track the amendments in their representation (Heer and Robertson 2007). It also has an aesthetic appeal, which is an important characteristic of the interface (Cawthon and Moere 2007).

This section describes the solutions, which were used in this research to support data analysis and processing.

3.3.1 QGIS as the main visualization environment

Most of the data in this research project are spatially distributed, and their visualization (or mapping) is a complicated, yet a common problem. Among the existing software tools that could handle this task for this research project, preference was given to an open source solution, QGIS (`http://qgis.org`). This application is a free geographic information system (GIS), developed and maintained by a community of volunteers around the world.

All user data in QGIS are arranged in a form of projects, which in turn consist of individual data layers (Figure 3.22). These layers are linked to their data sources and are drawn on a two-dimensional coordinate plane in the same order as they appear in the list. It is possible to add and remove layers and also to adjust their visual representations. Some visual parameters of objects (e.g. colour, opacity or thickness) can be linked to the properties of the underlying

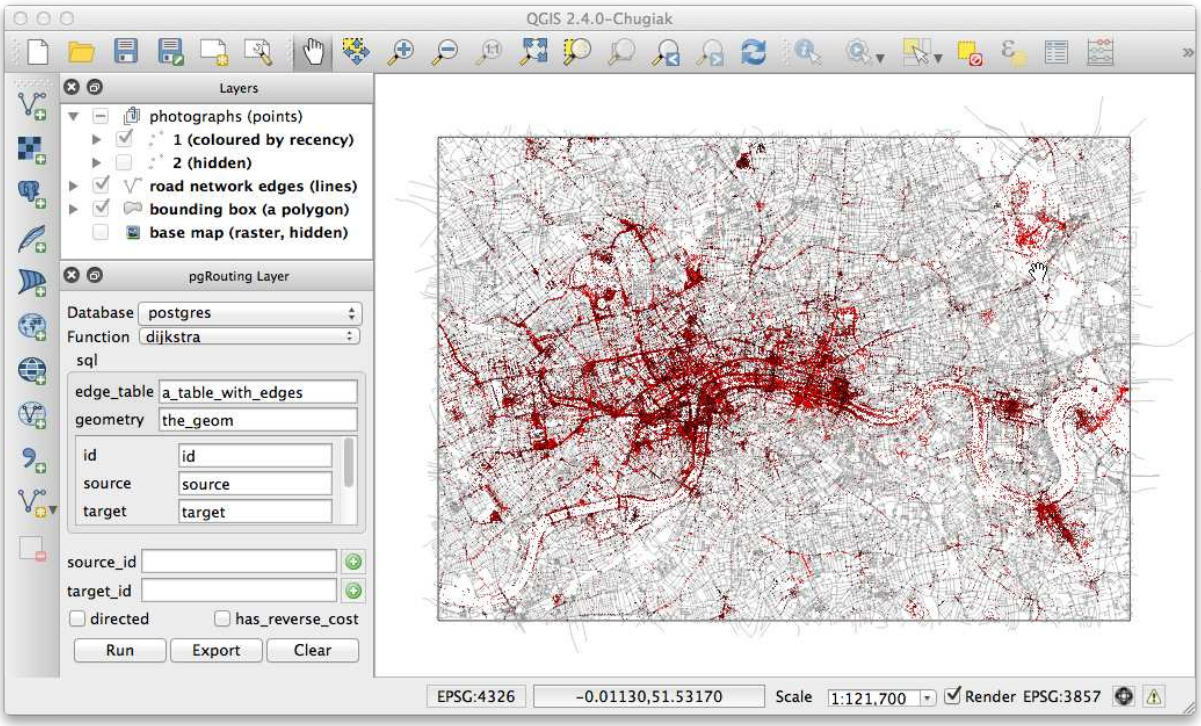


Figure 3.22: QGIS 2.4 showing sample layers of all required types and a ‘pgRouting’ panel.

records, which is a useful feature in the data analyses, allowing users to investigate spatial dependencies. There are a number of data source types supported by this software, including the stack of Postgres/PostGIS, used in the Dataset Abstraction Framework. QGIS can draw points, line strings and polygons, i.e. all geographical data types in this project. In addition, QGIS can be used to test and visualize some basic routing algorithms by means of a plugin called pgRouting (`http://plugins.qgis.org/plugins/pgRoutingLayer/`). This functionality was found helpful in this project as well (see Section 5.4 on page 246).

The above features and the overall accessibility of QGIS as a tool that did not require any software development, made this application the main data visualizing platform in this research. Most of the maps in the following chapters are also rendered with QGIS.

3.3.2 Interactive visualization of photographic datasets

Despite that QGIS was found a powerful instrument, some of its limitations made it necessary to develop two additional task-specific software tools for visualizing spatial data. The first one was needed for the exploratory analysis of the photographic collections.

As of version 2.4, QGIS does not store underlying data in memory (Dobias 2014), which results new requests to a database or a file system with any change in a viewspan. This works fine with hundreds or thousands of mapped features, but the performance significantly decreases as the sizes of the drawn datasets grow. When there are hundreds of thousands or millions of data records to show, every rendering takes several seconds, which makes real-time interactions hardly possible. As the exploration of the photographic datasets was one of the key stages in this research and the number of items to explore was expected to be over a million, the task could not be accomplished with existing software. Therefore, it was decided to implement a new application to serve this particular purpose.

The tool (Figure 3.23 on the next page) loads a configurable set of photographic collections from a Postgres database and shows a distribution of their locations on a map similar to QGIS. A user is able to zoom and pan the map, but because the data is kept in the memory, rendering is done much faster: it is instant at higher zoom levels and takes about one or two seconds when a viewspan includes millions of loaded records. Drawing is done in a separate thread,

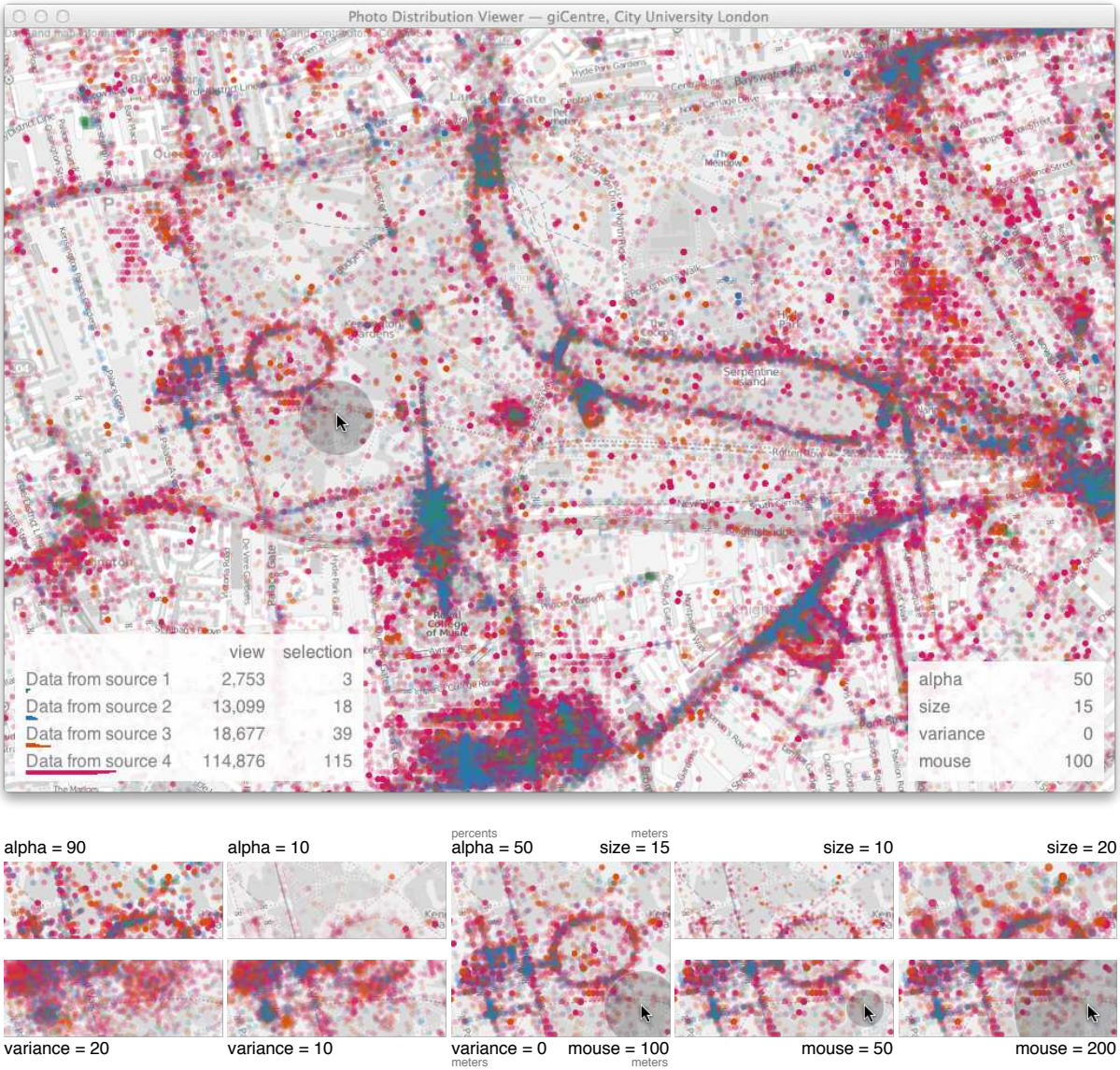


Figure 3.23: A tool for interactive exploration of photo distributions. *Top:* User interface, *bottom*: examples of visual parameters. The data on the screenshots are explained in Chapter 4.

so even if the data cannot be updated instantly, interaction commands are never interrupted or delayed. A user can control the visibility and the order of the loaded layers and also change some visual parameters of the rendering. These parameters include the transparency of dots (alpha), their size and 'variance'. 'Variance' in this context refers to a random component, which can be added to the coordinates of the photographs on demand. The higher the variance the more the deviation between the real and the drawn location of a photograph. Such feature supplements transparency in attempts to reveal anomalies in the distributions of images.

The bottom-left corner of a window shows some contextual statistics for the loaded datasets. A user sees the numbers of photographs in the view and also around a cursor. By moving a mouse over the map, it is possible to investigate local anomalies with no need to zoom in or change the ordering of layers. The proportions between the sizes of the datasets within a viewspan, around a cursor and in general are displayed as threefold bar charts under the titles of the loaded photographic collections. These bars also work as a colour legend, thus having dual functionality.

3.3.3 Visualization of quad-based data processing

Another scenario, when use of QGIS becomes difficult, implies dealing with datasets that are being updated in real time. As of version 2.4, in QGIS it is only possible to refresh existing layers either by pressing `F5` or by changing a viewport (i.e. dragging or zooming). Both methods are undesirable as they require user actions that carry no information Raskin (2000, p. 84). When the updates are needed with a small delay and over a long period of time, use of QGIS becomes impractical.

In this research, real-time tracking of changes is important when some data are being processed by quads (see pages 83-86). Such operations involve multiple parallel threads, which simultaneously update or populate a single dataset component. These complex procedures can be difficult to monitor and control for a number of reasons:

Each thread has its own independent state at every point in time. Some instances are busy processing quads, some are dividing existing quads into smaller ones, others are waiting for an empty free quad to handle it.

Performance rate depends on the features of the underlying data. Although it is always possible to compute an overall progress of an operation by estimating the proportion of areas that have already been processed, this information may not be sufficient. Some regions can require more computational effort due to spatial variations in data, and this slows down or accelerates the procedure over time as a whole. Sudden changes in performance may also be a result of problems in software, and a distinction between these cases can not be made automatically with ease.

Causes of errors can be hard to investigate. When data processing takes place under the conditions of uncertainty (e.g. when all peculiarities of the data are not known in advance), errors are rather likely. The nature of some of them can be related to spatial patterns in data, which means that they are more likely to occur at places with certain features. Finding a cause of an error can be challenging either when it is rare and 'random' or common, but unsystematic at first glance.

To overcome these difficulties and thus ensure the correctness of all quad-based procedures, it was decided to implement a separate standalone application that would provide real-time visual feedback about the progress of an arbitrary chosen operation. To make the tool independent from the nature of a task, it was made focused only on the quads (sectors) and their statuses, not the data that are being changed.

The interface of the software is shown in Figure 3.24 on the facing page. It mainly consists of a map, which shows all currently existing sectors in a chosen dataset component. Unprocessed sectors are transparent, and their fill colour changes to yellow when they are locked by a thread, green when ready and red in a case of a failure. If a thread is able to report a degree of completion for a locked sector, fill also works as a progress bar, expanding from left to right. The map can be zoomed and panned. The top-most part of the window contains a progress bar, which helps estimate the overall degree of task completion. The view is updated with the new data once in a second, which is on one hand enough for the purpose of monitoring and on the other hand does not overload the hardware.

An operator can hover over a sector and get its details; focusing on parent (divided) sectors is also possible (this is done by setting a layer offset value with a keyboard). Two process-controlling operations are available: incomplete sectors and those that contain errors can be reset, i.e. made unprocessed. This functionality is accessible via keyboard shortcuts as well and is useful in partially failed experiments (e.g. due to a network error).

The application works independently from a script, which does data processing, and can be launched or exited at any time. As all other implemented software in this project, the tool is accessible through the DAF's command line.

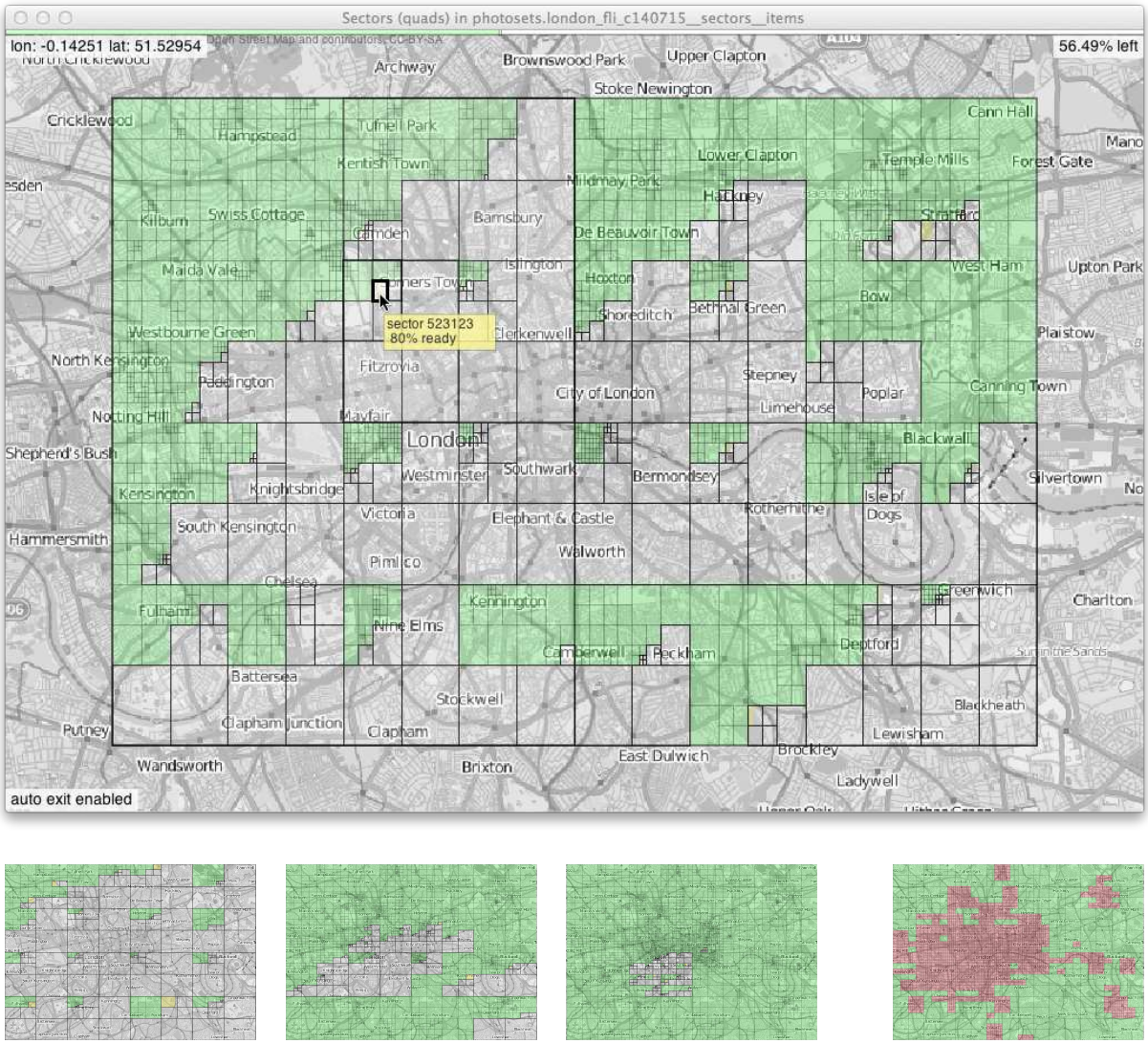


Figure 3.24: A tool for visualizing quad-based data processing. *Top:* User interface; *bottom-left:* different degrees of completion; *bottom-right:* an operation that has partially failed.

3.3.4 Exploration of survey results with glyphs

A survey on the contents of a sample of photographs, included into the workflow of this research (see page 43), was a source of another collection of information to analyse. The structure of these data is shown in Figure 3.25 on the next page. Each participant was shown a random sequence photographs (subjects) and left one or more responses, all consisting of a set of subjective classifications. The number of answers per response in this research was from

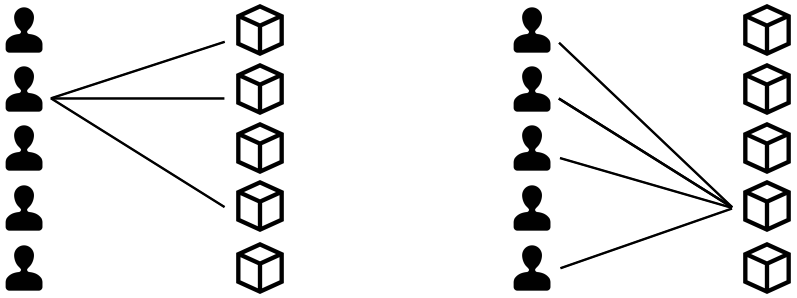


Figure 3.25: Symmetry of one-to-many relationships in a dataset containing crowd-sourced subject classifications. Each line is a survey response, consisting of several answers.

one to seven, according to the questionnaire description on page 46. Thus, every photograph was also related to a set of users, resulting a complex interlacement of links.

Survey data of similar kind are commonly visualized to summarise findings, while cleaning and processing are performed using statistical methods (Sapsford 2007). However, doing the analysis by means of visual exploration can be more beneficial in some cases because it can help find unknown hidden trends and better understand the dataset. This brings a need to effectively represent larger volumes of data to support decision making.

Commonly the results of the categorical surveys are aggregated into tables (Sapsford 2007) and are represented in a tabular form. Despite the simplicity of the approach it is rather ineffective for exploring the relationships in the samples and extracting key trends. To facilitate the comparison between the values belonging to different groups conditional formatting (Microsoft 2010; Abramovich and Sugden 2005), table-lens (Rao and Card 1994) or other techniques can be applied. While tables are highly informative and precise, they are not very suitable for exploring multi-dimensional survey datasets or unaggregated data.

The simplest and probably the most widely applied techniques of graphical representation of categorical survey data are bar charts and pie charts (e.g. Office for National Statistics 2011; O'Brien 2013). While being familiar to a wide audience and easy to understand, they are characterised by a low data-ink ratio (Tufte 1983) and thus are only suitable for visualizing highly aggregated data. Besides, pie charts are often criticised for causing difficulties in comparing proportions between categories and problems with scaling underlying values (Wilkinson and Wills 2005; Few 2007; Tufte 1983).

Often the structure of categorical responses allows more sophisticated and data-rich visualization techniques. For instance, spatially tagged survey results are laid over conventional geographical maps, displayed on Choropleth maps or structured in a form of spatial treemaps (United States Census Bureau 2013). With the emerging power of accessible and easy to use computer technologies different visualization techniques are more frequently combined together, giving an opportunity to see collected survey results in multiple views and to interact with them (Office for National Statistics 2011; L. S. R. Online 2010).

A special case of categorical responses is the Likert scale (Tastle and Wierman 2006), which is used when the categories are ordinal. This type of scaling is widely applied for rating subjective opinions and is also used in this research. To visualize the spread between the categories box plots can be used instead of bar charts or in conjunction with them (Salkind and Rasmussen 2007).

No evidence of use of high-resolution visualization techniques for exploring raw results in categorical surveys was found, and a new solution was designed to approach survey analysis.

When a survey comprises a set of subjects that are classified by respondents using a number of criteria, a single response may be described as an array of natural numbers. This array is a subset of a matrix with questions (criteria) in rows and answers (categories) in columns. A group of responses aggregates individual answers and can be presented as a table with their frequencies. Another characteristic may be introduced to show relationships between classifications: it counts frequencies of all possible matching pairs of answers. This may help to reveal common patterns in behaviour, which is useful for detecting insincerity (deliberate insertion of erroneous responses) (Amegashie 2007) and understanding opinions of participants about subjects. The total number of parameters describing a group increases exponentially as the answer space grows and can be reduced if only relationships between answers to neighbouring questions are considered. This still allows patterns in individual responses to be detected while excluded parameters may be brought back into the set by changing the order of questions.

The challenge is to represent all these parameters compactly in order to both evaluate them and to be able to visualize multiple groups of responses using juxtaposition for the purpose of comparison (Gleicher et al. 2011).

Inspired by the successful application of glyphs in a wide range of fields (Brandes and Nick 2011; Wickham et al. 2012; Maguire et al. 2012; Lie, Kehrer and Hauser 2009), it was decided to apply this technique to survey data analysis. A glyph "is a small visual object that can be used independently and constructively to depict attributes of a data record or the composition of a set of data records" (Borgo et al. 2012). A good overview of possible glyph designs can be found in Ward (2002) and Ward (2008). After considering a number of options, it was decided to construct a survey response glyph as shown in Figure 3.26:

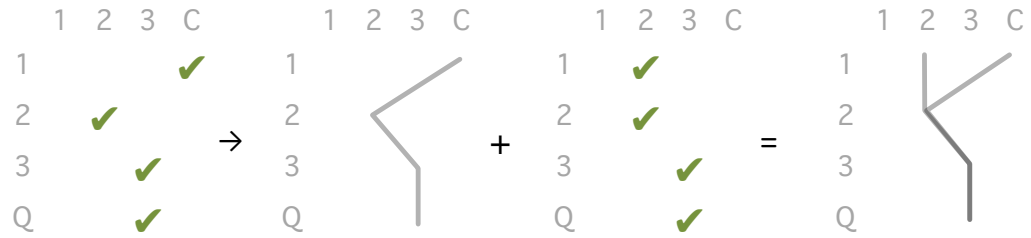


Figure 3.26: Response glyph concept. *Q* stands for survey *questions* and *C* denotes *categories*.

A glyph is similar to a parallel coordinate plot (Inselberg and Dimsdale 1990) rotated by 90°. A base for the glyph is a *survey response grid* having questions (dimensions) in rows and categories in columns. Each response is represented with a semitransparent polyline, thus both showing individual answers and links between neighbouring questions within a response. With this approach it is possible to show multiple response profiles in a single view by laying them on top of each other. Such representation allows to: (1) to see each individual answer, (2) to observe individual responses, (3) to visually estimate the number of answers in a group, (4) to see the most frequent answer profile and (5) to estimate the amount of disagreement among answers. Because such design is sensitive to ordering of categories (answers), it better fits ordinal types of variables, but can also be applied for nominal values or ordinal scales mixed with a nominal value (e.g. rating + not applicable).

Because the glyphs are distinguishable even at small sizes, it is possible to display a large number of them on a single screen to aid comparison. Laying them out as a grid with cells of 20 by 20 pixels in row-prime order, similar neighbours may be compared while also scanning the list vertically for broader scale changes in response patterns. The maximum number of glyphs that can be displayed on a screen depends on the resolution and the amount of space

left for other data views or interface elements. For example, given a 17″/1,280×1,024 screen, there is enough space for about 2,500–3,000 glyphs. Scroll bars are introduced if more space is required.

Having two lists of glyphs with responses grouped by survey participants and subjects, it is possible to see thousands of answers with very little level of aggregation and apply a visual analytic approach to analysis of these data. More details on the concept of a survey response glyph as well as the its application outside this research can be found in a publication presented at The Eurographics Conference on visualization in June 2014 (see Appendix D on page 300).

The interface of a tool built for response exploration using glyphs as well as the findings derived from the 'photo content assessment' survey data are described in Section 4.4 on page 176.

Chapter 4

Analysis of photographic datasets

Following the discussion of the research workflow in Section 3.1, this chapter focuses on a study of a set of photographic sources. It reflects the experiments, conducted to assess crowd-sourced collections of geotagged images as estimators of street attractiveness in urban areas. The order of the sections in this chapter repeats the sequence of steps, proposed on page 43. The target, towards which the chapter moves, consists of two components: (1) conclusions about how distant the chosen datasets are from a *model photographic collection*, defined on page 20 and (2) descriptions of filtering methods that can be applied in order to reduce existing discrepancies (details can be found on page 40).

4.1 Selection of the data sources

The proposed methodology does not orientate itself on any specific source of geotagged photographic data and can be applied to more than one collection of crowd-sourced images. Furthermore, candidate datasets do not have to meet the requirements that a *model photographic collection* corresponds to (their list can be found on page 38). However, there are two properties, the lack of which make the consideration of a photographic source impractical. First, it needs to be popular and well-known, because the whole idea of measuring street attractiveness via the spatial density of photographs is in relying on the opinion of a crowd of photographers.

Second, a photo-sharing service needs to support geotagging and also provide legal means to extract existing images at a given location. Without this feature, it is impossible to obtain a local copy of a distribution of photographs and thus calculate the attractiveness scores.

Taking into account previously conducted studies and the information on photo-sharing websites available online, in this research it was decided to consider the following image sources:

Flickr, `http://www.flickr.com/`
This is a general-purpose image and video hositng website, allowing users to share their photographs both privately and publicly. The service has been launched in 2004 and is owned by Yahoo since 2005. Flickr data are available to software developers via an API (Application Programming Interface), which has got a rich collection of methods to receive or upload photographic information. Flickr has been considered in numerous research projects, including those that have been mentioned in Chapter 2 (e.g. Arase et al. 2010; Jain, Seufert and Bedathur 2010; Alivand and Hochmair 2013).

Geograph, `http://www.geograph.org.uk/`
According to the official website, Geograph is a project that aims to collect spatially representative photographs and information for every square kilometre of Great Britain and Ireland. Initiated in 2005, this photo-hosting website contains over four million photographs taken by more than twelve thousand people across the two counties. Although the service does not cover the entire planet, Geograph is available for London and hence has been found to be potentially useful for measuring street attractiveness in cities on the British Isles. There are two ways of data data harvesting from this source: (1) with an API and (2) directly, via a bulk download of the whole database. Geograph appears in research projects not as frequently as Flickr, but still receives some attention (e.g. mySociety 2009; Purves, Edwardes and Wood 2011).

Panoramio, `http://www.panoramio.com/`
Panoramio is a global community-powered site for "exploring places through photography". It works since 2005 and is currently owned by Google. Every uploaded image is a candidate to be shown in Google Earth and Google Maps, so as in Geograph, there exists a set of rules the photographers are encouraged to follow. All images pass

human-made moderation, usually within one or two days after being uploaded. Although the database with images cannot be accessed directly, the information on photographs at a given location can be gathered via an API. Just as Flickr, Panoramio is commonly included into the studies of crowd-sourced spatially distributed data (e.g. Kisilevich et al. 2010; Yin et al. 2012; Zielstra and Hochmair 2013).

Picasa Web, `http://picasaweb.google.com/`
This project is another photo-sharing website, owned by Google. Being one of the earliest crowd-sourced image collections (the launch was in 2002), Picasa Web is commonly compared to Flickr for similarity – the photographs are also not moderated. The popularity of this service is partially caused by the existence of a desktop photo-managing application with the same name, which allows users to share their albums online with almost no effort. Examples of research projects dealing with the data from this website are Antoniou, Morley and Haklay (2010), Anca-Livia et al. (2012) and Kurashima et al. (2012). For simplicity, this source of photographic data is referred as *Picasa* throughout this report.

Picasa was discontinued as a web service in late 2013, and the photographs were migrated to Google+ (Google 2013).

Other popular photo-sharing websites (Alexa 2014), such as Imugr (`http://imugr.com`), PhotoBucket (`http://photobucket.com/`), Instagram (`http://instagram.com/`), SmugMug (`http://www.smugmug.com/`), etc. were either found as not supporting georeferencing in general or were not providing third-party developers with means to harvest existing data.

4.2 Data gathering

In order to analyse patterns in the chosen photographic sources and propose filtering methods, it was necessary to cache the information about the images in the region, chosen in Subsection 3.1.3 (i.e. Central London). According to the workflow, introduced in Subsection 3.1.1 on page 43, different stages of the study required different kinds of data.

First, it was only necessary to work with spatial and temporal coordinates of all photographs, which would be enough to asses their distribution and suggest filtering methods of *type a* (the term is explained on page 40). Successfully filtered datasets and datasets with no anomalies were to be sampled and added to a *photo content assessment* survey. Depending on the results of the manually conducted classification, filtering methods of *type b* were to be proposed. As these filters are based on the metadata of a photograph, an image file itself or both, more information would be needed.

Taking this state of things into account, it was found beneficial to decouple the process of data gathering into these three general parts:

Distribution forming (compulsory). A local database (cache) is populated with records, representing all photographs that were taken within a chosen region. The data include longitude, latitude, the time of a shot and the name of a photographer, but may also be supplemented with some metadata, if this does not add complexity to the process (e.g. if a photo service API returns extra attributes in response to a basic search query).

Additional metadata gathering (optional). If some filtering methods require metadata, which are available, but have not been cached yet, this gap is filled. The process does not change the number of items in the cache, it only adds new attributes to existing records.

Image files gathering (optional). In cases when some filtering methods involve image processing, the photographs themselves need to be cached in order to be analysed by the corresponding algorithms. If the process is launched after all metadata-based filtering, a good amount of resources can be saved – rejected photographs can be skipped and do not need to be downloaded (a description of the approach to filter chaining can be found on page 41).

The process of *distribution forming* is reasonably straightforward when a photographic service provides direct access to the original database – no data can be lost during gathering. Searching for images by means of an API creates a bottleneck between the origin and the cache, which makes the latter potentially sensitive to errors and limitations. Consequently, a local copy of a distribution of photographs can obtain additional bias, which may create more barriers for the data to be used as a good estimator of street attractiveness.

This section describes the approaches, which were used to gather data from all four chosen sources, and also pays attention to some important issues.

4.2.1 Process description

With Dataset Abstraction Framework, introduced in Section 3.2, it was possible to work with different image sources in a common way, thus eliminating the difficulties in data gathering and processing. As all photographic datasets were similar in their nature, it was decided to adhere the same general data structure, which is shown in Figure 4.1.

All primary data except for the photographs themselves were stored in DAF datasets, belonging to domain `photosets`. Image metadata was distributed between two components: `items` and `item_details`. The first component was populated during the *dataset forming* and contained spatiotemporal coordinates, photo and user identifiers and also some metadata, which could be harvested together with the above attributes. Besides, this dataset component stored filtering flags (binary variables to indicate what filters each photograph satisfies) and all necessary derived attributes. The results of *additional metadata gathering* were put into component called `item_details`, where records shared their ids with `items`. This segregation of an single semantic entity (a photograph) contributed to robustness, which was important during the experiments, conducted under the conditions of uncertainty. *Gathered image files* were cached

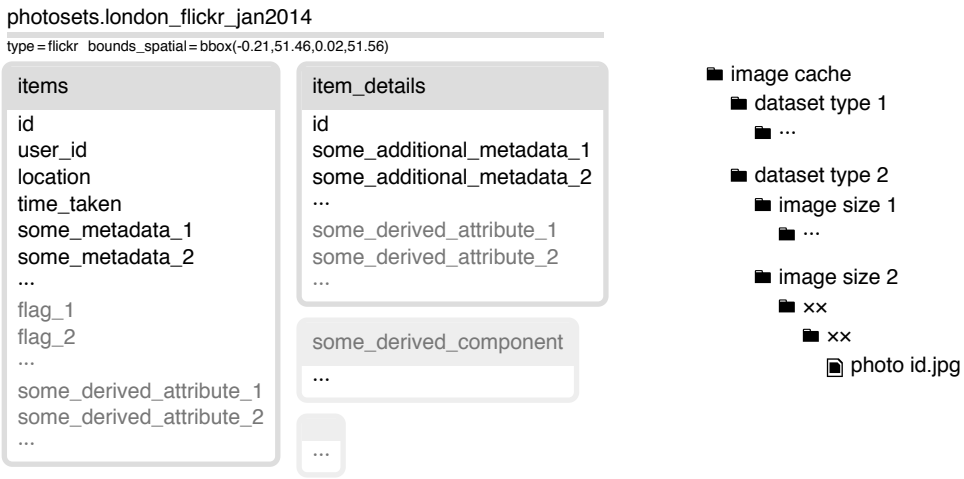


Figure 4.1: General structure of a cache for the photographic data.
Left: an example of a DAF dataset; *right:* image cache directory format.

to a local storage drive and were grouped by the types of DAF datasets (i.e. the origin of photographs) rather than the dataset names. This allowed multiple versions of the same dataset to share image files, which helped keep the size of the cache small, but could not negatively affect any of the experiments. Two extra levels of folder hierarchy were introduced in addition to `type` and `size`. They served a technical need to avoid directories with extremely large amounts of files, as this slows some types of filesystems (Trautman and Mostek 2000). The names of these directories equaled to the first and the second pair of characters from an md5 hash (Rivest 1992) of a photo id: `md5(123456) = e1 0a dc 39 49 ba 59 ...` Such approach limited the number of subfolders at each level to 256 ($\texttt{00-ff}_{\texttt{hex}}$), thus dividing the collections of files to up to 65,536 parts.

To gather all primary data for each photographic dataset in this research it was necessary (1) to initialise it, (2) to assign it with properties such as type and spatial boundaries, (3) to populate component `items`, (4) to populate component `item_details` (if it is was relevant to datasets of the chosen type), and finally (5) to cache images for a subset of records (only those that passed all metadata-based filters).

With the command-line interface, suggested by DAF, these steps could be easily performed the following way for any of four data sources and for any arbitrary geographic region:

```
$ cd photo-routing
# 1 - initialise of a new dataset, which belongs to domain 'photosets'
$ app/console daf:datasets:init photosets.london_flickr_jan2014

# 2 - assign properties to a new dataset
$ app/console daf:datasets:components:properties:set photosets.london_flickr_jan2014
    type flickr
$ app/console daf:datasets:components:properties:set photosets.london_flickr_jan2014
    bounds_spatial "bbox(-0.21,51.46,0.02,51.56)"

# 3 - populate component 'items' with records (distribution forming)
$ app/console daf:datasets:components:records:populate photosets.london_flickr_jan2014
    items
```

```
# 4 - populate component 'item_details' with records (additional metadata gathering)
$ app/console daf:datasets:components:records:populate photosets.london_flickr_jan2014
    item_details
```

```
# 5 - cache images for a subset of items (image files gathering)
$ app/console pr:photosets:images:download london_flickr_jan2014 640
    --filter="flag_xxx=true"
# this command is application-specific and therefore has namespace 'pr' (for
    photo-routing) rather than 'daf' (for Dataset Abstraction Framework)
# 640 is a desired image with
# flag_xxx is a boolean attribute, equal to 'true' for those records that have passed
    all metadata-based filters
```

The behaviour of commands 3, 4 and 5 depended on the `type` of the given dataset.

Geograph was the only source that provided direct access to the original database with all attributes of all existing photographs. **Distribution forming** in this case was straightforward, and no records could be lost during this process. Component `items` was populated from a local copy of a database, which contained all four million records. The creation of this auxiliary database was also automated with DAF and consisted of the follosing two commands:

```
$ app/console pr:photosources:geograph:download-snapshot /geograph-snapshot-dir
Downloading category_canonical...
[============================] 100% 10115 Bytes
Downloading category_stat...
[============================] 100% 167277 Bytes
Downloading gridimage_base...
[============================] 100% 124523132 Bytes
#...
$ app/console pr:photosources:geograph:restore-db-from-snapshot /geograph-snapshot-dir
Creating database geograph... Done.
Importing category_canonical.mysql.gz into the database... Done.
Importing category_stat.mysql.gz into the database... Done.
Importing gridimage_base.mysql.gz into the database... Done.
#...
```

Flickr, Panoramio and Picasa do not provide the developers with an opportunity to obtain all photographic metadata at a chosen location just as Geograph, but allow them to perform image search via publicly available APIs (Application Programming Interfaces). It is possible to submit the coordinates of a region of interest to a server and in return receive a list of photographs, reported by users as taken within it. The regions in all three cases have to be rectangular and their shape is defined with four numbers: minimum and maximum longitudes and latitudes in WGS84. API-based search methods do not guarantee to respond with a full list of all images that exist in the original database, and there are differences in the limitations that they set up:

> **Flickr** – `https://www.flickr.com/services/api/flickr.photos.search.html`
> In order to be able to search for photographs by a rectangular bounding box in Flickr, it is necessary to define what is called a *limiting agent* "in order to prevent the database from crying" – otherwise only images taken within the last 12 hours are included in the response. Additional parameters such as `min_date_taken` or `min_date_uploaded` can be applied here. Search can be also refined with the accuracy of a geotag, which varies from 1 (world) to 16 (street level / default). It is possible to adjust a set of attributes

the API returns and in addition to basic image properties receive textual description of a photograph, owner name, tags, etc.

Apart from a list with image metadata the response always includes the total number of items that satisfy the given search criteria. If this quantity exceeds the maximum number of photographs per query (100 by default, 500 maximum), it is possible to run additional queries with the same criteria and parameter `page = 2, 3, etc.` to access more image data. According to the design of the Flickr's search method, there is no upper limit for the overall size of a list, so even if the chosen region is large and the API reports millions of photographs, one can scan through the whole range of thousands of pages. However, the actual size of a multi-page list neever exceeds 4,000 unique items. According to a comment from a member of Flickr team, this constraint is conditioned by a need to optimise the work of the servers (Flickr 2007).

To control the load of the API, the service requires the developers to sign their requests with an API key and run not more than around 3,600 queries per hour on average. Access to the API can be withdrawn if abuse is detected.

In this research it was chosen to use `min_date_taken = 2000-01-01` as a limiting agent (this date is four years before the launch of the service). Search was also narrowed down to items with `accuracy = 16` as other photographs were less likely to be relevant to the places where they are tagged (Hauff 2013).

Panoramio – `http://www.panoramio.com/api/data/api.html`

This service provides two types of APIs: *Widget API* to display photographs on a website and *Data API* to obtain metadata of existing images as an array of objects for various purposes. Search in Panoramio Data API is simpler than one in Flickr and supports only two parameters in addition to the coordinates of a geographic region: `set` and `size`. Parameter `set` can be equal to `public` (popular photos), `full` (all photos) or contain a numeric user id. For `size` it is allowed to use `original`, `medium` (default), `small`, `square` and `mini-square`. This parameter does not affect search results and only changes attribute `photo_file_url` in the response (i.e. a link to an image file).

Panoramio API supports pagination and allows developers to retrieve more items than it can be contained in a single query (i.e. 100). Documentation does not clarify if there is any limit to the number of unique items a sequence of paginated responses can return,

but there is evidence from some API users that it exists (e.g. Elotheos 2013; Sizo 2013). API requests do not have to be signed, but it is still not recommended to perform more than 100,000 requests per day to avoid fees.
In alignment to the methodology of this research it was chosen to use `full` as a value for `set` in all queries to Panoramio (i.e. to collect as much existing photographs as possible).

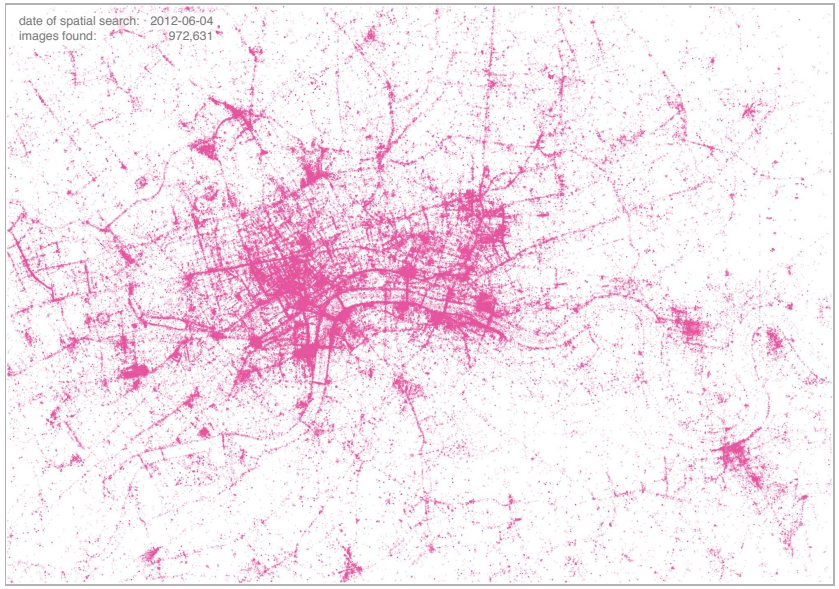
Picasa – `https://developers.google.com/picasa-web/docs/2.0/reference`
The latest version of Picasa API provides access to user data in a form of customisable feeds, which can include privately shared photographs and albums if the requests are authenticated and authorised. Public search accepts `bbox` as what is called a 'response filter', which makes it possible to gather metadata of images taken in a chosen rectangular region. The maximum size of a feed, defined by the protocol, is 1000, i.e. two pages with up to 500 records in each.

Because of the limitations in all three APIs it was not possible to form representative distributions of photographs simply by running search for the entire chosen region – the datasets would be incomplete. To overcome this problem it was decided to use the quad-based data processing approach, inspired by Tchaikin (2008). A general description of this method is given in Subsection 3.2.4 on page 83 and Subsection 3.3.3 on page 99.

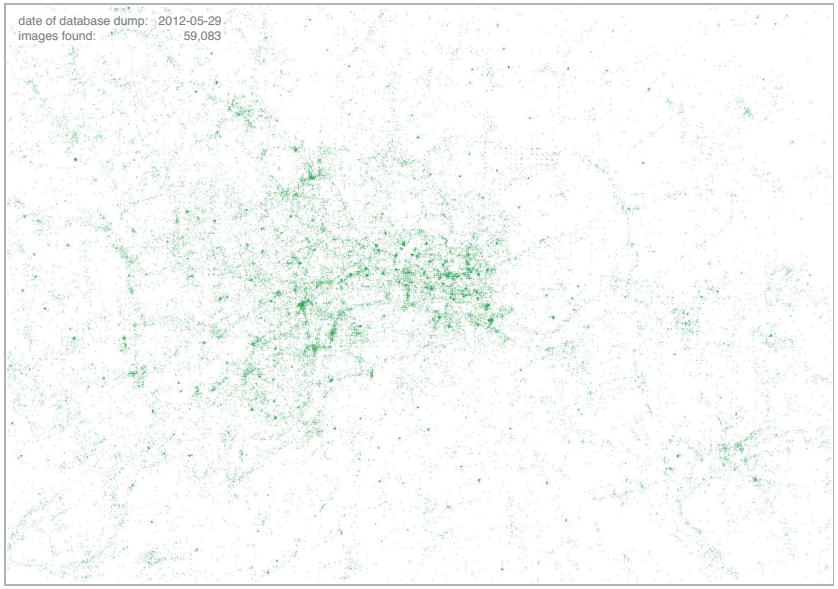
Each of the three data gathering scripts was making a spatial search request to a corresponding API and if a reported number of found images was more than a threshold, it divided a region into quads until this was no longer needed or when the minimum allowed region size was reached. The higher the density of data in some area was, the more resources were required to cache the existing photographic distribution, but the smaller the chance of record loss became. Unpopular areas among photographers could be still processed quickly.

The first distributions of images, which were obtained in 2012, and are shown in Figure 4.2 on the following page. A look at them in the photo distribution viewer (Subsection 3.3.2 on page 97) confirmed the existence of irregularities that were described in previous research and could be used as an instrument for measuring street attractiveness. None of the photographic sources were rejected at this stage.

(a) Flickr



(b) Geograph



(c) Panoramio

(d) Picasa

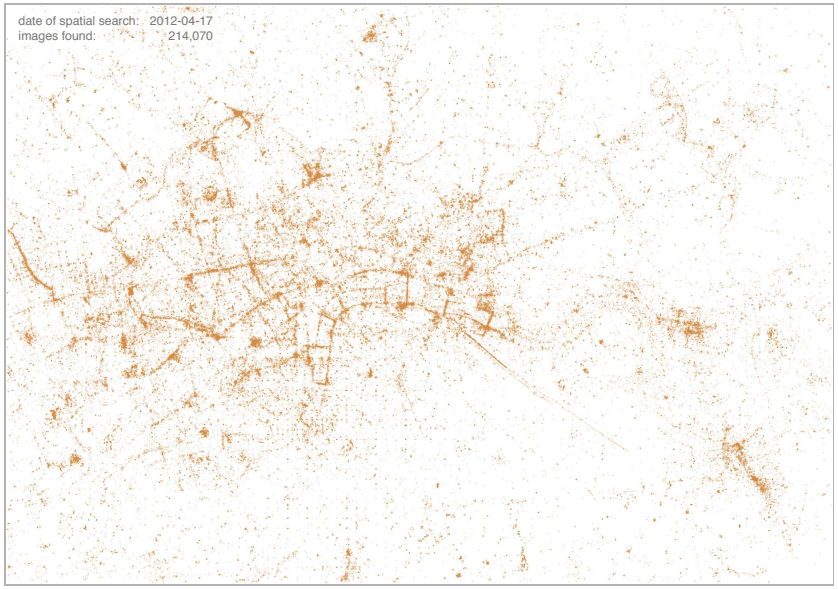


Figure 4.2: The first results of *distribution forming*.

Additional metadata gathering (the next part of data caching) was helpful in investigating more parameters of photographs. This operation could be useful for Flickr, Panoramio and Picasa, because search functions in the APIs of these services were not designed to return all known information about the images, such as EXIF data etc. EXIF stands for *Exchangeable image file format* (Camera & Imaging Products Association 2010) and is widely applied to store camera settings, date and time of a shot, its GPS coordinates and custom user data. Some, but not all of these properties are automatically extracted by photo-sharing websites to become available in search results (e.g. spatiotemporal coordinates).

A set of scripts was written to gather additional metadata. These programs looked at the records in component `items` of a chosen dataset, and if there were ids that did not have a pair in component `item_details`, individual requests to photo-sharing services were made and `item_details` was populated with what was received in return. Additional metadata from Flickr was cached by means of individual API queries (`https://www.flickr.com/services/api/flickr.photos.getExif.html`). The same task for Panoramio and Picasa could be accomplished by crawling web pages with image descriptions. A common shell for all three scripts supported multiple parallel threads, which significantly increased performance.

A summary of attributes, available for caching, is shown in Table 4.1 on the following page.

Image files gathering (the third part of data caching) was needed to facilitate potential filtering methods that deal with the contents of photographs.

Because shared user-generated data can be used in different circumstances (e.g. on desktop computers and mobile devices, on dedicated pages or in albums), photo-sharing services make the image files available in several sizes. A summary of all discovered options is shown in Table 4.2 on the next page. The bigger images are used in the analyses, the more accurate results can be obtained, but the more disk space and computational resources are required. As an optimal value for image size could not be predicted without running the experiments, photo gathering scripts were made able to cache all versions of existing photographs.

To cache image files from Flickr, Panoramio and Picasa it was only necessary to read property `image_url` from component `items`, replace characters that encoded the size and then simply download a file from the resulting web address. Image URLs (uniform resource locators)

	Flickr	Geograph	Panoramio	Picasa
photo id	•	•	•	•
location (lat, lon)	•	•	•	•
location precision	•	•		
author id and name	•	•	•	•
time of photographing	•	• (date only)	○	•
time of sharing	•	• (date only)	•	•
title	•	•	•	•
description	•	•	○	•
tags	•	•	•	○
page url	•	•	•	•
image url	•	○	•	•
EXIF	○		○	○
moderation category		•	○	
annotated faces	○			
view count	•			
license type	•		○	

Table 4.1: Image attributes available for caching from the selected photo-sharing services.
• – could be gathered during *dataset forming*;
○ – required additional requests to the servers.

	Flickr	Geograph	Panoramio	Picasa
100	•		•	•
120		•		•
213		•		•
240	•		•	•
320	•			•
500	•		•	•
640	•	•		•
800	after 2012-01-03			•
1024	after 2010-05-25		•	•
> 1024	original	original	original	any (dynamic)

Table 4.2: Sizes (longest sides) of image files available at the selected photo-sharing services.

in Geograph were supplemented with a secret hash, which was introduced by the developers to avoid unfair use of direct links to photographs (hotlinking) – this could abuse the servers if somebody embedded original images to a very popular internet resource. To overcome this issue, image gathering script opened web pages with the descriptions of photographs (their addresses were known) and obtained the hash from there.

4.2.2 Issues and ways of resolving them

As it was noted earlier, all three chosen photo service APIs did not guarantee reliability and completeness of cached photographic datasets by their design. Lack of attention to the details of the process of data gathering could result unwanted bias in the distributions of images and thus negatively affect the correctness of conclusions in this research.

Although some API-related issues were mentioned in previous studies (e.g. in Hietanen, Athukorala and Salovaara 2011), it was necessary to carefully investigate them from scratch and suggest ways of avoidance if possible. Problems in data gathering were detected by means of visual analytics and by tracing errors in corresponding scripts. Some issues were having permanent nature, but some emerged or disappeared over time with changes in server-side software (the data in this project were updated multiple times between 2012 and 2014). A summary of detected problems is provided below. The list includes only issues related to photographic data gathering – the anomalies in the original spatial distributions of photographs are discussed later, in Subsection 4.3.3 on page 157.

Flickr: circular regions instead of bounding boxes

The most fragile part of image data gathering was *dataset formation*. Any unnoticed concealed problem in this process could seriously impact all other stages of this research. Taking into account that spatial search functions in all three chosen photo-service APIs did not guarantee the integrity of the provided data, it was important to detect factors, which negatively influenced derivable distributions of images. An example of such a factor was a temporal issue in Flickr API, which emerged in 2013, but was later fixed in spring 2014. In spite of the statements in the documentation, search by a bounding box returned photographs outside the defined regions, forming circles around them (see examples in Figure 4.3 on the next page). This issue

Figure 4.3: Examples of Flickr API returning records for circular regions rather than specified bounding boxes.

could be easily left unnoticed without visual means to assess the results of data crawling, and looked minor ot first glance. However, it could cause the exclusion of significant volumes of photographs from the API responses, taking into account a limitation of maximum 4,000 records per a search request. As the quad-processor was making a decision to split or process a given sector based on the reported existing image count, it could start collecting records in large, but relatively unpopular areas, which were located near small, but very popular ones. API responses would be dominated by the records outside the chosen regions, thus making it impossible to access many photographs that were expected to be returned. The more stretched a rectangular areas was, the more it became vulnerable to this issue, because the ratio between the size of its own surface and the size of the circumscribed circle decreased.

This problem was reported to Flickr developers (see `https://groups.yahoo.com/neo/groups/yws-flickr/conversations/messages/8675`) and was permanently fixed in spring 2014. As a temporal solution for its eliminating, the maximum sizes of quads were decreased and a *split/process* threshold was lowered at times when it existed.

Flickr: no photographs near prime meridian

The second Flickr API issue was also discovered in 2013. It was easier to detect, but incorrigible. After the developers updated the internals of the API, all photographs with longitude between -0.0001 and 0.0001 (excluding zero) disappeared from the results of spatial search (see Figure 4.4 on the facing page). The problem was first observed in an updated version of the cached Flickr distribution and was later confirmed in several experiments. The same problem was also found relevant to areas near equator, where latitude is approaching zero.

Figure 4.4: A problem with gathering photo metadata from Flickr near prime coordinates. *Left:* Greenwich park, London, UK (prime meridian); *right:* Middle of the World park, Pichincha, Ecuador (equator).

Blanks in a spatial distribution of 'votes' such as the described ones guarantee low or null attractiveness scores for some pathways even if they are popular among the photographers. A source of data cannot be considered as reliable for certain urban areas if they contain such problematic places and there are no ways to overcome the cause of the issue.

The problem with the prime meridian in Flickr API was not fixed until the final round of data collection in this research, so to keep using Flickr, records with longitude between –0.0001 and 0.0001 were borrowed from an older dataset, collected in summer 2013.

A discussion of this problem can be found in the official Flickr Developer Support Group: `https://groups.yahoo.com/neo/groups/yws-flickr/conversations/topics/8786`.

Flickr: errors in coordinate parsing

Another issue related to the prime meridian and Flickr was found in early 2014. During several attempts to update a distribution of photographs from this source, small quads near longitude of zero started taking unexpectedly long time to proceed (Figure 4.5a on the next page), and the sizes of the new cached datasets grew enormously.

Investigation showed that this strange behaviour was caused by the API, which started failing in parsing numbers in scientific notation. If any of the edges of a bounding box contained a small number, e.g. `0.00001234`, the value was automatically translated to `1.234E-5` by the client software, and Flickr servers treated it incorrectly, just as numbers without the exponential part. This made the chosen region 'spill', so that the distribution of photographs started occupying thousands of square kilometers (Figure 4.5b on the following page). Although the

(a) Quads near longitude of zero remain unprocessed even when all other regions are complete.

(b) Boundaries with numbers in scientific notation force regions to 'spill'. The radius of the distribution here is approximately 350 km.

Figure 4.5: Number formatting issue in Flickr API, resulting vast amounts of photographs outside the chosen bounding box appear in the cached versions of image distributions.

problem was fixed by Flickr developers in Summer 2014, this case suggested to check all client libraries and add more control to the process of API request formation in order to avoid similar problems with other data sources in future.

Panoramio: discontinuity of the spatial distribution in the most popular areas

Visual exploration of the locations of photographs in several versions of Panoramio data revealed what was later called 'virtual edges' at two locations (Figure 4.6 on the next page). These edges (or discontinuities) looked like an artefact of data gathering rather than patterns in the original distribution. Changes in the sizes and the positions of the quads did not affect the locations of these artefacts. The comparison of two datasets, which were gathered with one year interval (Figure 4.7 on the facing page), suggested that this issue could be related to the way Panoramio structures the data before making it available via the API. It was speculated that the servers probably split the photographs into 'tiles', which are then queried individually according to the API requests. If the tiles can store not more than a predefined number of records, areas around extremely popular venues (such as The Houses of Parliament or the Tower of London) become underrepresented. This peculiarity of Panoramio API potentially makes the attractiveness scores for some streets smaller than they should be, which is a sign of unsuitability of this source of data for the chosen purpose. However, this problem was considered as not severe, because the affected streets were nevertheless surrounded by significant numbers of 'votes'.

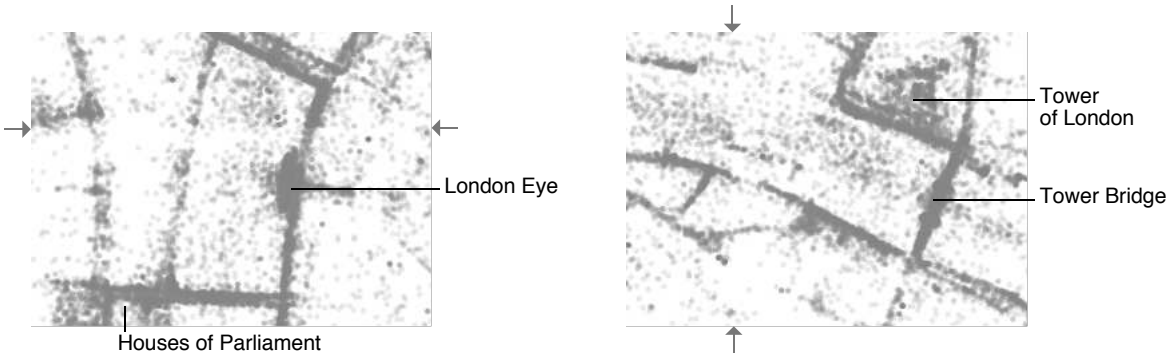


Figure 4.6: 'Virtual edges' in spatial distribution of Panoramio photographs.

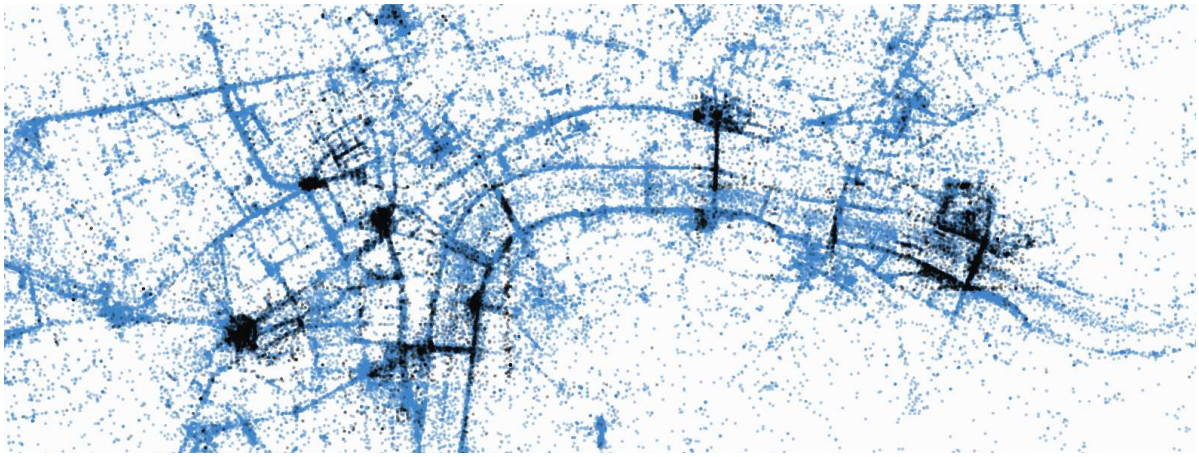


Figure 4.7: Spatial distribution of Panoramio photographs that were deleted by owners or excluded from API responses within one year period. Black dots represent records that could be collected in Summer 2012, but were no longer found by the crawler in Summer 2013. A few rectangular patterns with clear edges emerge near the most popular venues.

Panoramio: masked API failure

Apart from the problem described above, some early versions of Panoramio data were also having randomly arranged gaps in various parts of the chosen region (Figure 4.8 on the next page). These blanks were not coherent with any failures in the data gathering software, but were clear signs of a problem. After a number of experiments it became clear that the issue was caused by a rare fault in the Panoramio API, which was in reporting no images in places where they actually existed. The error was masked behind a valid API response and appeared to have a random nature – no influencing factors could be determined. Luckily, it was easy to overcome such a problem by repeating suspiciously served requests: in the majority of cases just a single consecutive query was enough.

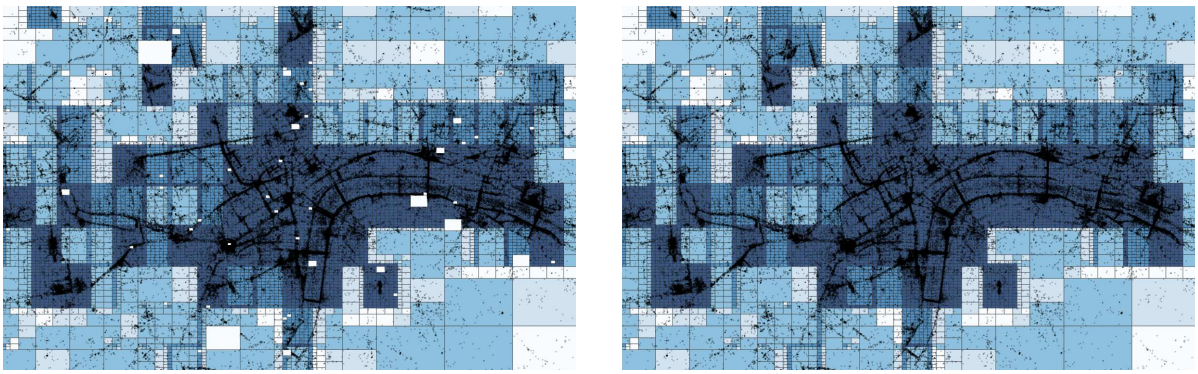


Figure 4.8: Masked API failures in Panoramio: some quads are empty in places where they are expected to contain images. Sectors are coloured by the total number of existing images, reported by the API. *Left*: no attempts were made to recollect the data. *Right:* the crawler requested the contents of empty quads up to three times.

Picasa: impossibility to retrieve most of the existing records in popular areas

Similarly to Flickr and Panoramio, spatial search in Picasa API could only return a limited number of records for a requested bounding box (1,000). However, the quad-based approach to *distribution forming* for this source of photographs could not help overcome this constraint in full. Exploratory analysis of gathered Picasa data revealed that some popular areas only contained the most recent images and were becoming empty as the maximum value of *time of photographing* attribute was decreased in visualizations (Figure 4.9). Use of smaller *split/process* thresholds or the reduction of the minimum quad size did not change the situation.

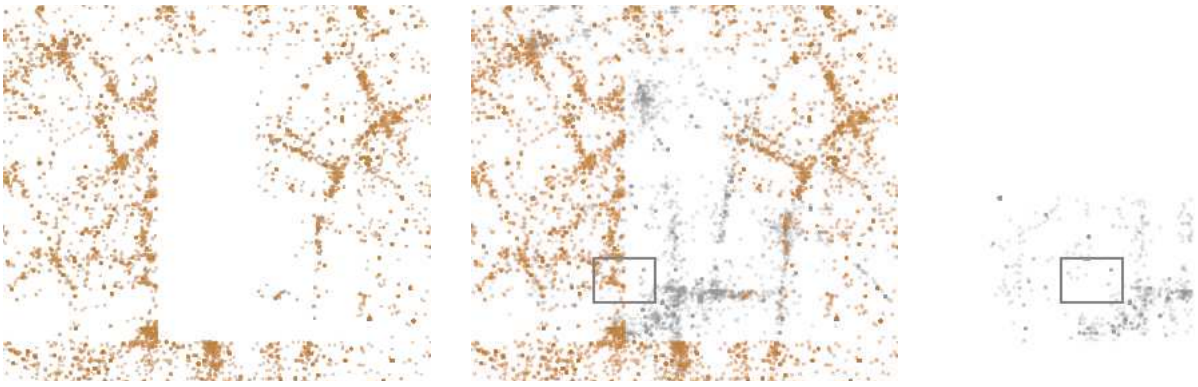


Figure 4.9: Incorrigible issue in spatial search of Picasa API – the majority of records in popular areas cannot be retrieved. *Left:* Spatial distribution of photographs in Westminster, obtained in summer 2012 and showing only photographs taken before December 31st, 2011. *Middle:* The same dataset with all records. *Right:* Results of a single search query.

The cause of the problem was found through visualizing results of individual API requests. It became clear, than no matter how small a requested bounding box was, records in API responses were spread across larger regions, approximately 0.5 km by 0.5 km in size. Because API servers ordered images by recency, it was impossible to know, for example, how photographers behaved near the Houses of Parliament only few months ago. This fact, combined with an API limit of 1,000 metadata entries per spatial search, made it impossible to gather a representative distribution of photographs for many parts of Central London. Hence, it was decided to reject Picasa as a potential source of 'votes' to form street attractiveness scores, because no other publicly available means of data gathering could be applied for this photo-sharing service.

Picasa: major API outage

The previously described unavoidable problem was not related to Picasa data as such, so positive changes in its API could return this photographic source to the study. A change did occur in late 2013 when this service was merged into Google+ (Google 2013), but the effect was opposite. Figure 4.10 shows spatial distributions of two Picasa datasets, collected before and after this event. As it can be seen from the images, either Picasa API was considered as redundant and was discontinued, or all photographs except a few exceptions were removed from Picasa when migrating to Google+. In any case, the 'votes' from this service became unusable even in places where the numbers of photographs in the smallest served bounding box was less than one thousand.

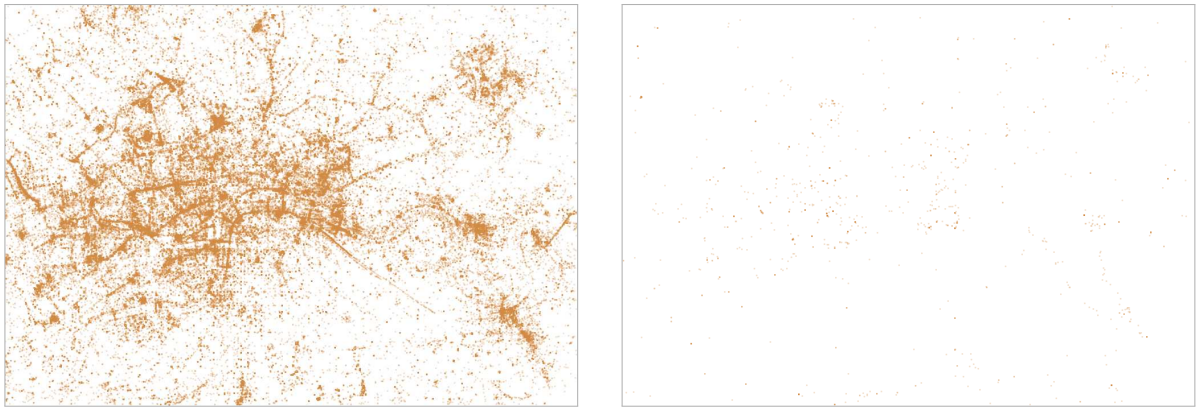


Figure 4.10: Results of changes in the internals of Picasa API in late 2013. *Left:* A distribution of photographs, returned by the API in July 2013 (209,297 records). *Right:* A distribution of photographs, returned by the API in July 2014 (2,151 records).

Flickr, Panoramio, Picasa: unavailability of additional metadata or image files

Separation of data gathering into three parts (*distribution forming*, *additional metadata gathering* and *image files gathering*) was a base for supplementary freedom in experiments, but at the same time potentially led to difficulties in maintaining data integrity. The further the parts of this process were apart from each other in time, the higher the chance of item disappearance at the origin became. This could happen mainly because some images were deleted by their owners or made private – changes in the results of spatial search did not affect neither the availability of additional metadata nor the pictures themselves, because both were accessed via direct links.

In real photo-based pedestrian routing systems, where bias-reduction filters are known in advance, all necessary parts of data gathering can be done all together. However, even in these circumstances the metadata for some images may be still found partially unavailable. Frequency of such cases and their reasons are discussed later in this chapter.

All services: unavailability of recently taken photographs

The final issue of data gathering to mention is not related to technical aspects as such and concerns more general principles of photo-sharing. After a person has taken a picture at some interesting location, thus 'voting' for its attractiveness, this information remains hidden from others until a photograph is uploaded to the photo-sharing website and added to the results of spatial search. Absence of significant proportions of 'votes' from certain time periods can potentially lead to bias in attractiveness scores, especially if some filters involve spatiotemporal clustering or consider seasonal bias. Hence, it was decided to look at how long it took the photographs to appear in publicly available distributions.

The experiments have shown that Flickr and Picasa APIs update search results in lass than 24 hours, and the same process takes about a week in Panoramio. Geograph database dumps are updated on daily basis. Differences between *date of photographing* and *date of sharing* are shown in Figure 4.11 on the facing page. The graphs demonstrate that most of the images are put online not right after they are taken, and the time intervals after which they appear on photo-sharing websites are different. Flickr appears to be the most dynamic service (about 40% of items are shared on the same day), while Panoramio users tend to delay with uploading their works more than elsewhere. In all four cases spatial distributions of photographs from

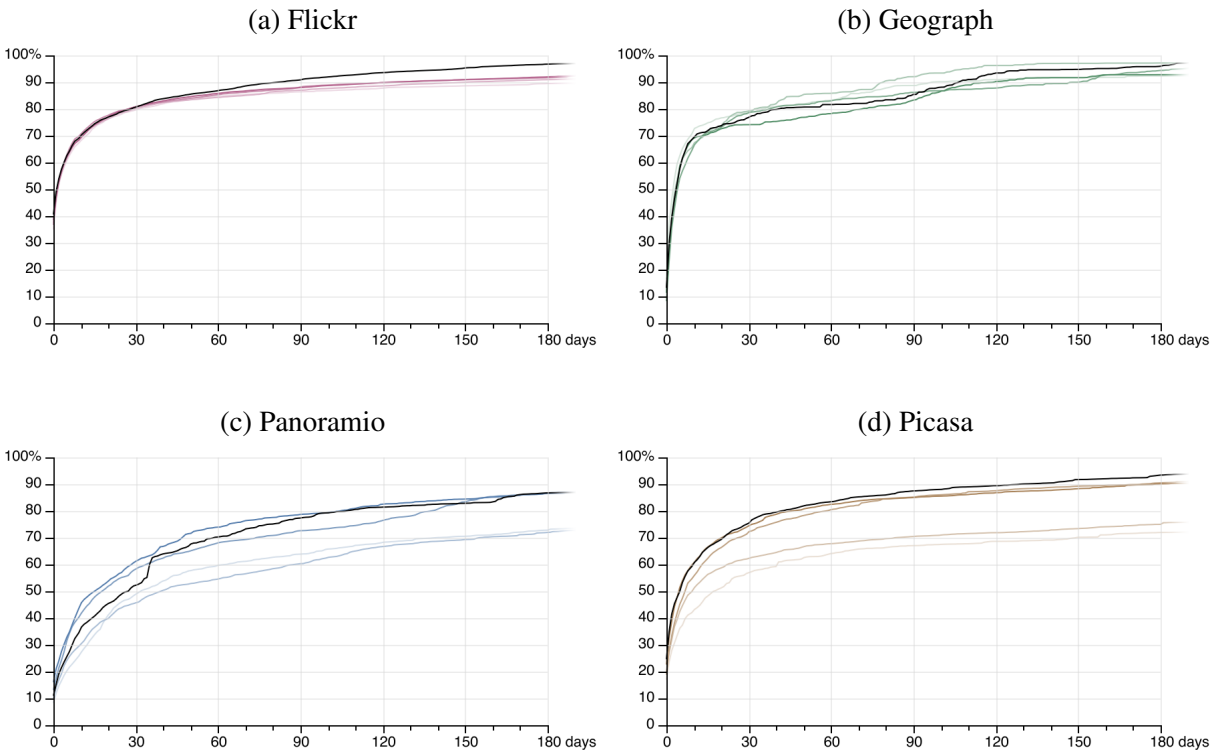


Figure 4.11: Time gaps between the image *date of photographing* and *date of sharing* by years. Values show the percentage of photographs with time gaps less than or equal to the given number of days. The darker the colour, the more recent the year. Black: the latest considered year (2012 for Picasa and 2013 for three other datasets). Data were collected six months after the end of the given period.

the most recent months remain incomplete, which can potentially introduce additional bias in attractiveness scores. To avoid this possible issue it was decided not to work with images taken within the last few months before data gathering. For simplicity the threshold was set to the end of the previous year or to the middle of the current year, depending on which date was more than one or two months away from the *distribution forming* date.

4.2.3 Adjustment of spatial search

At the beginning of this research project, when photo data gathering software was under development, there was an opportunity to check spatial search against another method of *distribution forming*. Cached Flickr records could be compared with a dataset kindly provided by

Professor Gennady Andrienko from Fraunhofer Institute, Germany (Peca et al. 2011). Their approach was in scanning user photostreams, thus gaining access to images across the whole world, including ones with no geographical coordinates. An sample from that database covering Central London, was merged with the cached Flickr distribution, which was obtained using spatial search. Items that did not present in both datasets were marked and then were loaded into the Photo Distribution Viewer to investigate patterns in their locations (Figure 4.12). The result was cleaned from the recently taken photographs to remove natural inequalities, caused by different *distribution forming* dates.

Exploratory data analysis helped understand that spatial search was able to return higher numbers of photographs in general, but at the same time was less effective for gathering records in the most popular places. Thus, the 'votes' were more dispersed in space than in the case of user photostream scanning, making attractiveness scores near the top landmarks not as high as they could be if there was an opportunity to work with all shared photographs. Such drawback of this data gathering method, however, could not negatively affect the quality of the generated routes, because the most attractive places still remained the most dense, so were the first ones to be assigned the smallest possible weighted costs by the routing algorithm (Equation 3.2 on page 54). On the other hand, a lack of photographs in less popular areas, which was observed in the second dataset, potentially resulted smaller attractiveness scores across the whole road network, and this effect was making values more vulnerable to noise. Thus, although it appeared that user photostream scanning was providing a more balanced distribution of images, spatial search remained more appropriate for the chosen purpose.

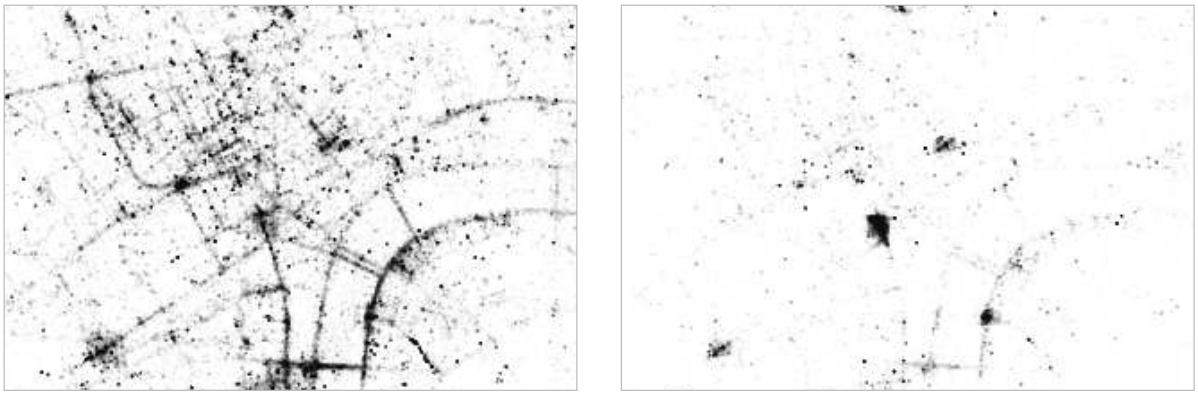


Figure 4.12: Locations of photographs from Flickr, which could be gathered using spatial search, but not by scanning user photostreams (*left*) and vice versa (*right*).

The above short study suggested to investigate how the adjustments in quad-processing approach to spatial search could influence the completeness of Flickr, Panoramio and Picasa datasets. The limitations on the numbers of unique records, which the APIs of these services were able return, meant that the smaller quads were queried, the more complete the distributions of photographs become. However, excessive divisions of sectors could only lead to extra unnecessary requests and not help harvest more data – the decisions to split the quads were becoming unpractical after reaching some point.

With Dataset Abstraction Framework (Section 3.2) it was easy to gather multiple collections of images in an arbitrary region using different sizes of quads. Thus, the most effective way of *distribution forming* for each source of photographic data could be established. Apart from measuring of the total numbers of sectors and photographs, this experiment was making it possible to investigate how the changes in software configuration affected the proportions of cached records at various locations. The area, chosen for this study, is shown in Figure 4.13. This 8×8 km square was selected for being rather diversified, including both popular tourist attractions along the Thames and quiet residential areas in southern parts of London.

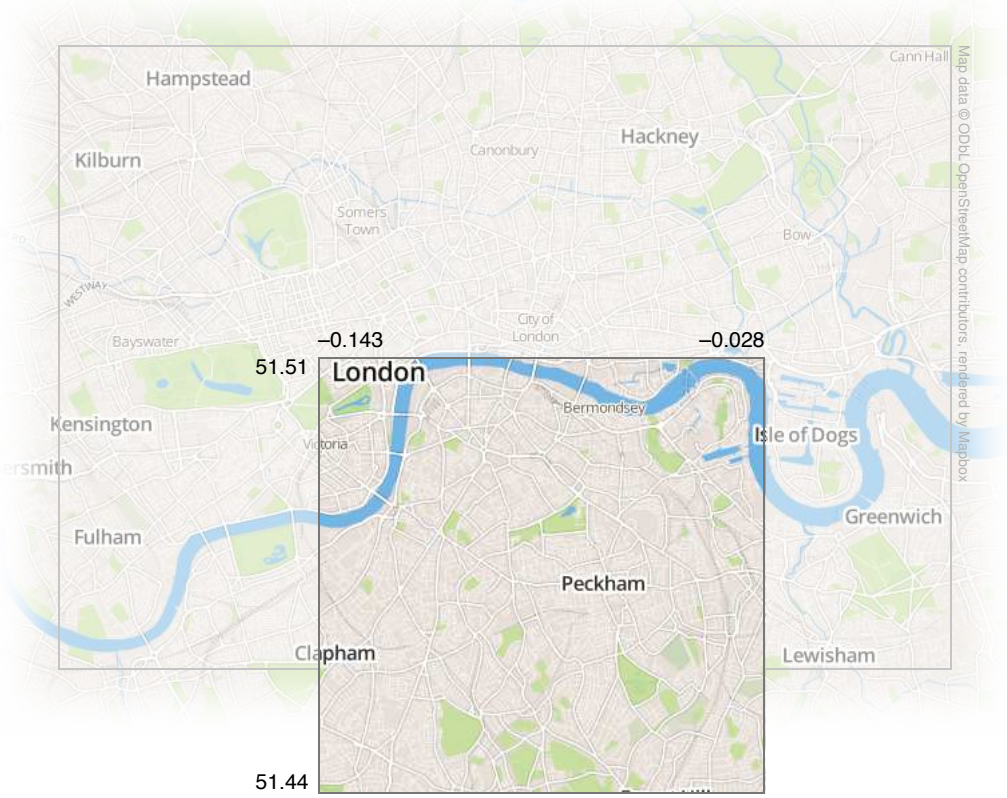


Figure 4.13: Bounding box of an area, chosen to test the effects of spatial search adjustment.

It was decided to test eight edge lengths for quads, making each two times smaller than the previous and thus varying sector sizes between 8×8 km and 62.5×62.5 m ($62.5 = \frac{8{,}000}{2^7}$). As each additional step increased the maximum numbers of quads in four times, extending the experiment to edge lengths of 31.25 m or even 15.625 m was considered as impractical (the total number of sectors to query would be up to 65,536 and 262,144, respectively). The goal of the study was to answer the following questions:

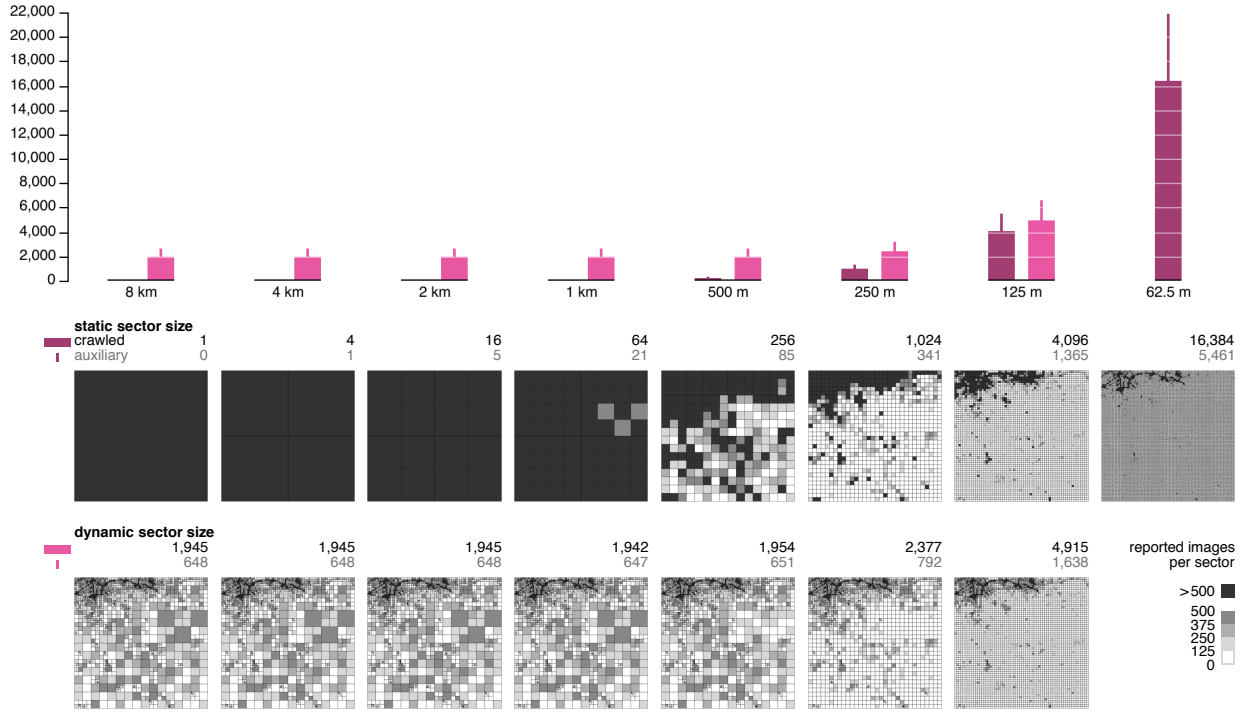
1. How many more photographs can be gathered with each additional sector division?
2. When do marginal costs surpass marginal gain? When to stop quad splitting?
3. How big can be the difference between the minimum and the maximum edge size when sectors are divided dynamically based on numbers of reported images in API responses?
4. How representative are the spatial distributions of images in different cases?

Lists of samples for each source of photographic data consisted of two arrays. The items in the first one were configured to have static sector areas, and the datasets in the second array were divided dynamically, differing by maximum sector size. *Split/process* threshold was set to 1,000 items in all cases (this number is less or equal to the documented limits for all three APIs). As it was impossible to obtain the original distributions of photographs for checking cached records against some 'ground truth', datasets were compared to the one with the highest number of quads (62.5×62.5 m, static sector size).

The results of this experiment are shown in Figures 4.14, 4.15 and 4.16 on three next pages.

Because the study was conducted after major changes in Picasa (Google 2013), it was only able to confirm the outage of its API, which was described on page 125. The results for Flickr and Panoramio were rather similar and carried no surprises. As expected, each extra sector division step was adding to the size of a gathered dataset, but marginal gain began to reduce when the lengths of the sectors were made smaller than 250 / 125 meters. The cost of using a static approach to quad handling was increasing exponentially, which was making it impractical for small sector sizes. At the same time, the results of dynamic quad processing were almost completely independent from a chosen maximum size of a quad, while the approach itself made it possible to significantly reduce the numbers of queries to the servers. Expectancy

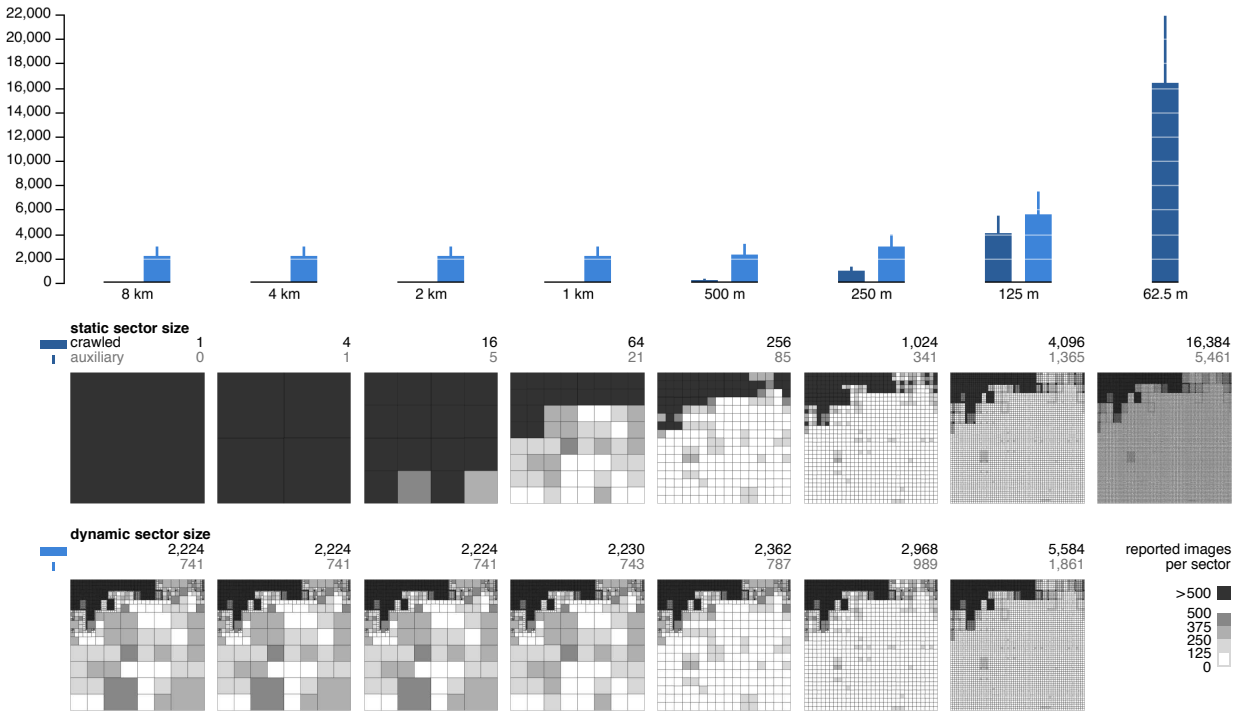
(a) Numbers of sectors involved and reported record count per sector



(b) Numbers of records gathered and the quality of their distribution

Figure 4.14: Adjustment of spatial search for Flickr. Each column represents the results of data crawling in two modes: with dynamic and static sector sizes. The bigger the pairwise difference between the lighter and the darker bars in *a*, the 'cheaper' it is to use static sectors of a given size. However, the bigger this difference in *b*, the less photographs are returned by the API when the static sector size is used. The more intensive the colour of the map cells in *b*, the more biased the spatial distribution of the resulting dataset (compared to what could be potentially retrieved from the API).

(a) Numbers of sectors involved and reported record count per sector



(b) Numbers of records gathered and the quality of their distribution

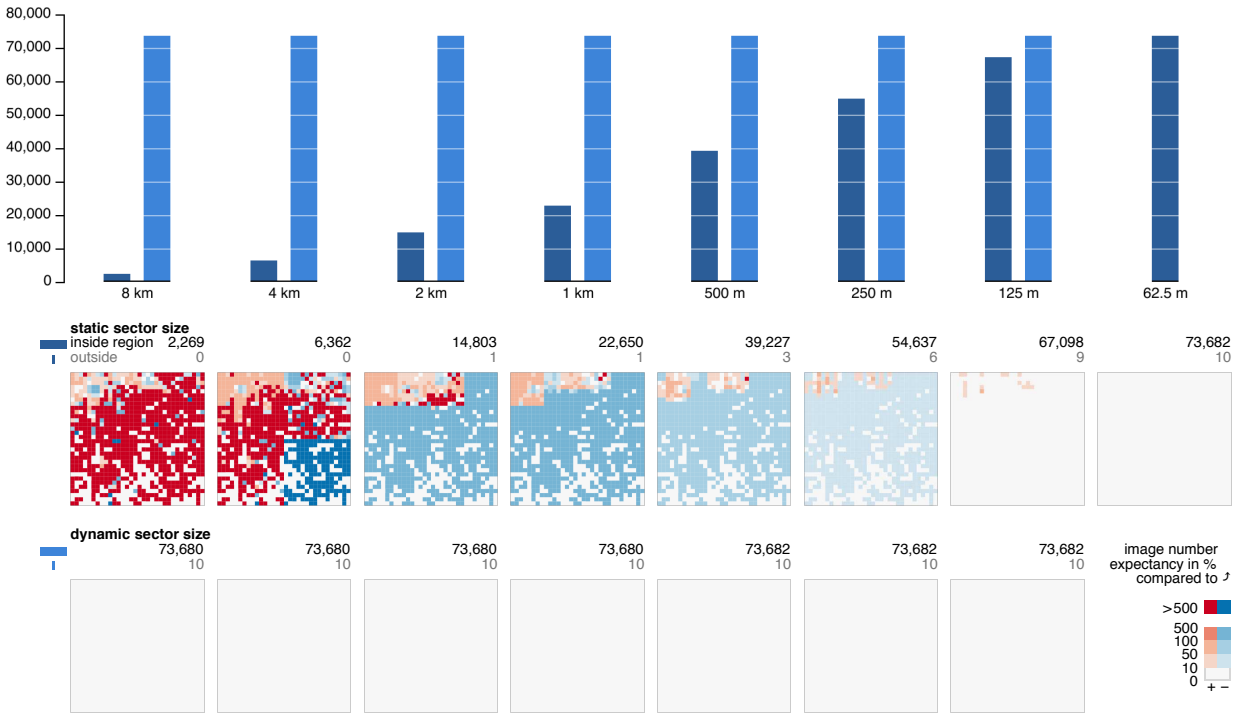
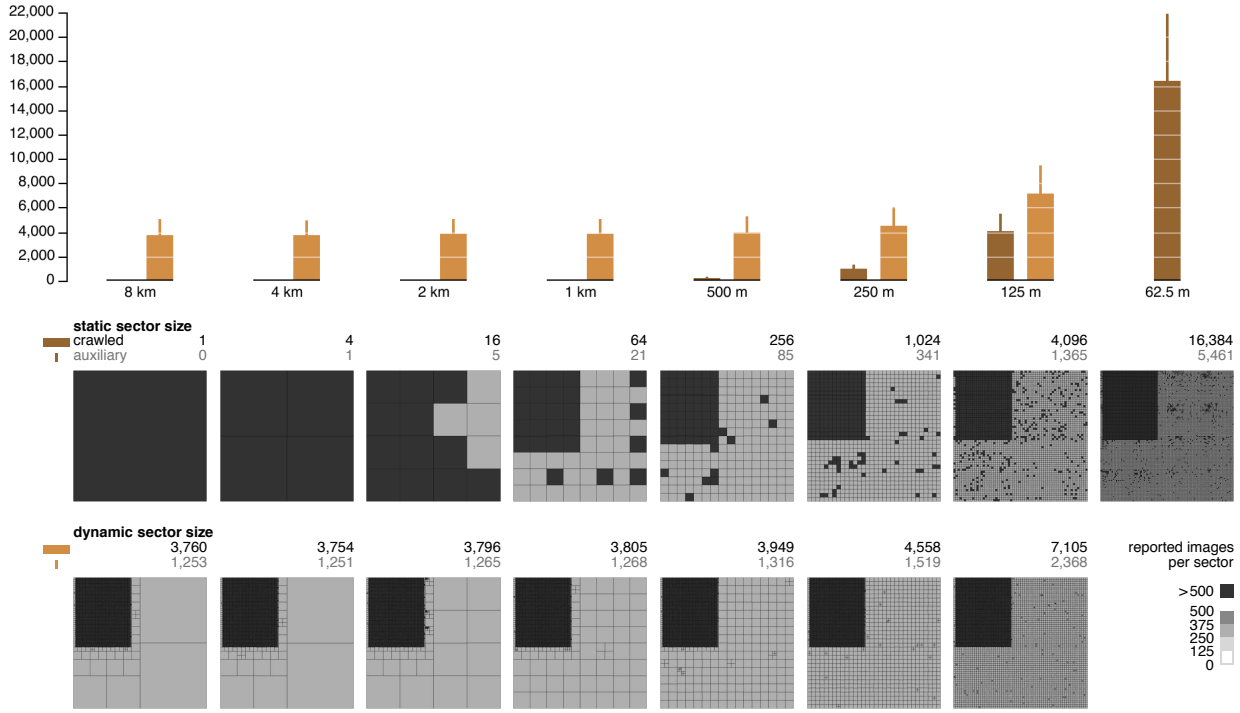


Figure 4.15: Adjustment of spatial search for Panoramio. See caption to Figure 4.14 on the preceding page for the details on this visualization.

(a) Numbers of sectors involved and reported record count per sector



(b) Numbers of records gathered and the quality of their distribution

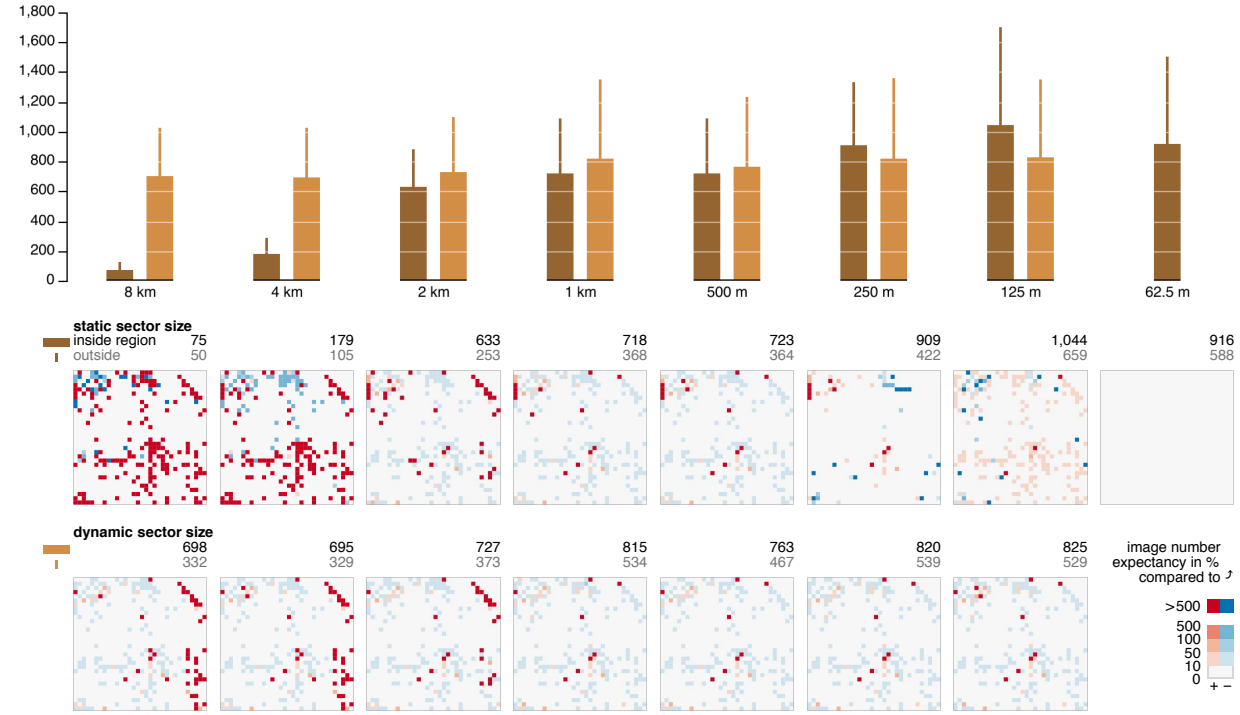


Figure 4.16: Attempts to adjust spatial search for Picasa. See caption to Figure 4.14 on page 131 for the details on this visualization.

maps, displaying the proportions of photographs across the chosen region in comparison to the reference dataset illustrated the danger of using larger minimum quad sizes both for Flickr and Panoramio. The fewer sector divisions were made in popular areas, the less represented these places became. As a consequence, attractive street segments could be getting smaller scores, and this would make the routing algorithm give them a significantly smaller preference.

Taking the result of this experiment into account, it was chosen to keep using dynamic quads in spatial search, not to set any limit for maximum quad size and divide sectors until they are approximately 100×100 m (aspect ratio may vary based on a shape of the root region).

A by-product of this study was a discovery of an unexpected artefact in Panoramio and Picasa APIs, which was later called 'tiling effect'. Instead of returning item counts for the requested bounding boxes, these two services were found supplementing the lists of records with another value, which represented numbers of photographs in fixed larger regions (tiles). The dimensions of these regions were not depending on precise size and location of a requested bounding box. However, if a quad appeared between the tiles, returned `item count` became equal to a sum of two or more values for neighbouring tiles. Besides, some Picasa responses were car-

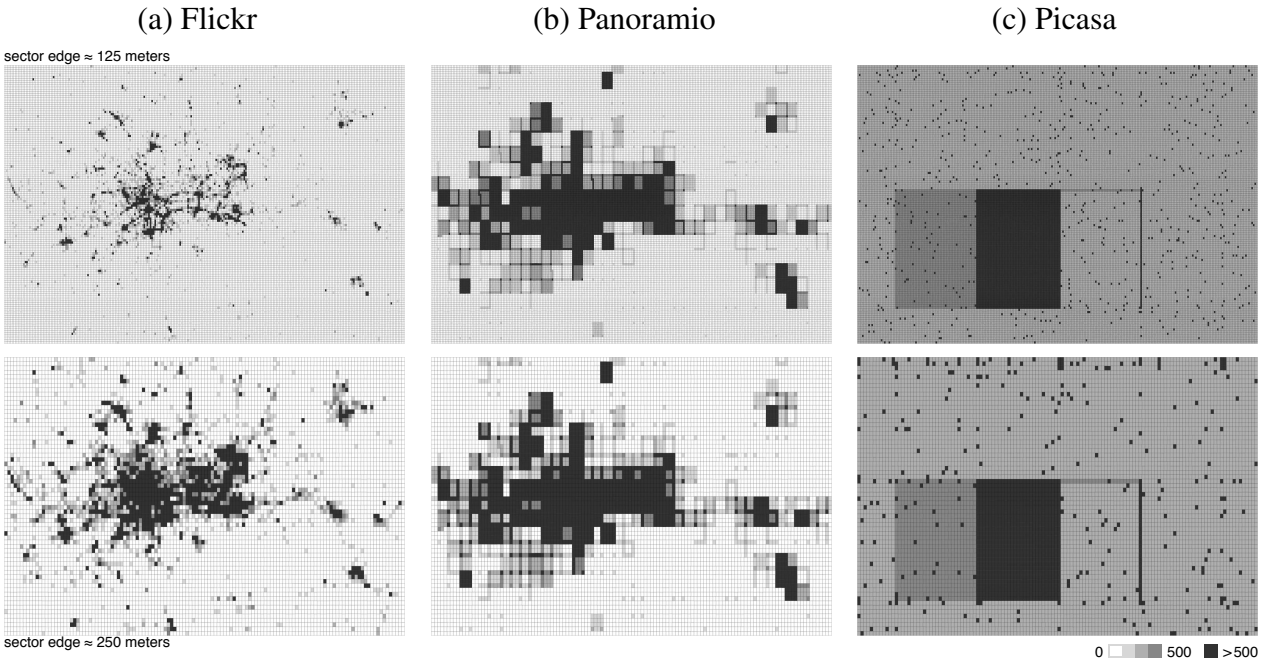


Figure 4.17: Numbers of photographs reported by service APIs in Central London and the dependency of this value on the sector size. 'Tiling effect' is observed in Panoramio and Picasa.

rying noise, i.e. the values were from time to time equal to unreasonably huge numbers (e.g. 15,365,481 or 9,153,783).

Perhaps, the best explanation of 'tiling effect' could be a need to optimize the process of serving search requests. This peculiarity of APIs could not negatively influence the resulting cached distributions of images, but was making the process of *distribution forming* less optimised. When `item count` per tile was bigger than a *split/process* threshold, quad-processor had to divide the entire tile into the smallest possible quads, thus making redundant requests near the most densely photographed areas. The scale of this shortcoming can be observed in recent charts on pages 131 and 132: given 8×8 km dynamic grid and a *split/process* threshold of 1,000, Panoramio quad-processor had to gather data for 2,224 sectors to get metadata of 73,680 photographs, while Flickr was dealing with only 1,945 sectors and obtained 799,186 records.

An example of 'tiling effect' is shown in Figure 4.17 on the preceding page; it can be also noted in Figure 4.8 on page 124.

4.2.4 Results

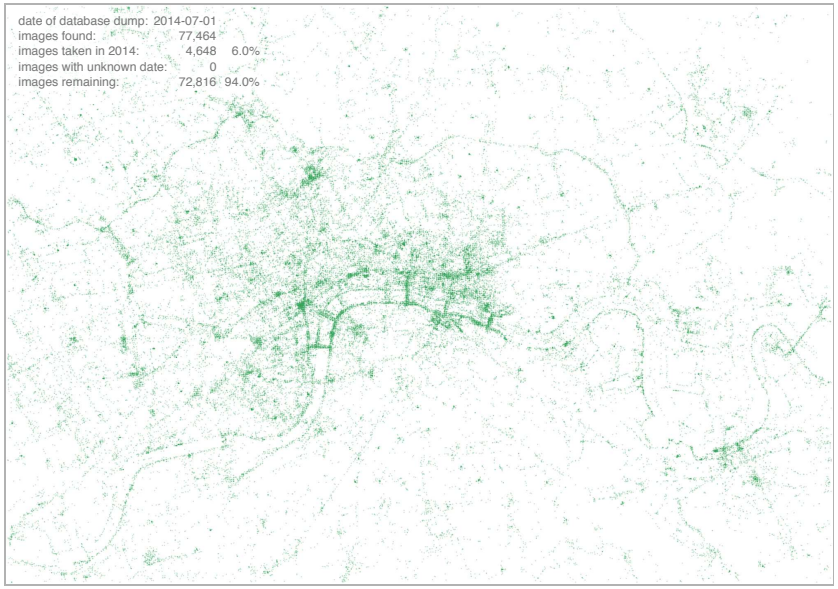
Detailed examination of the image data gathering process was an important step towards the creation of a reliable photo-based pedestrian routing system. Without ensuring that cached datasets were representative, it was easy to obtain a significantly biased spatial distributions of 'votes', and this could either facilitate inaccuracy in street attractiveness scores or lead to incorrect conclusions about the suitability of the chosen data sources.

This research dealt with several versions of image distributions from all four photographic services, as it was easy to repeat any data processing operation with DAF. To keep this report consistent, all Flickr, Geograph and Panoramio data mentioned onwards are from July 1, 2014 with pre-filtering of images taken after December 31, 2013. Picasa dataset, which was rejected for two reasons discussed earlier, appears in some charts as harvested on July 23, 2013.

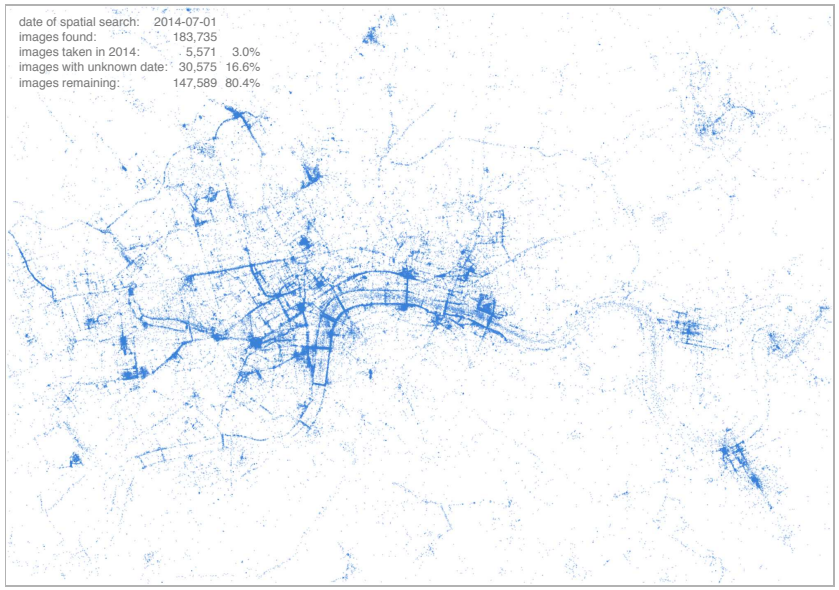
Spatial distributions of photographs as well as the sizes of the latest datasets can be found in Figure 4.18 on the next page.

(a) Flickr

(b) Geograph



(c) Panoramio



(d) Picasa (rejected)

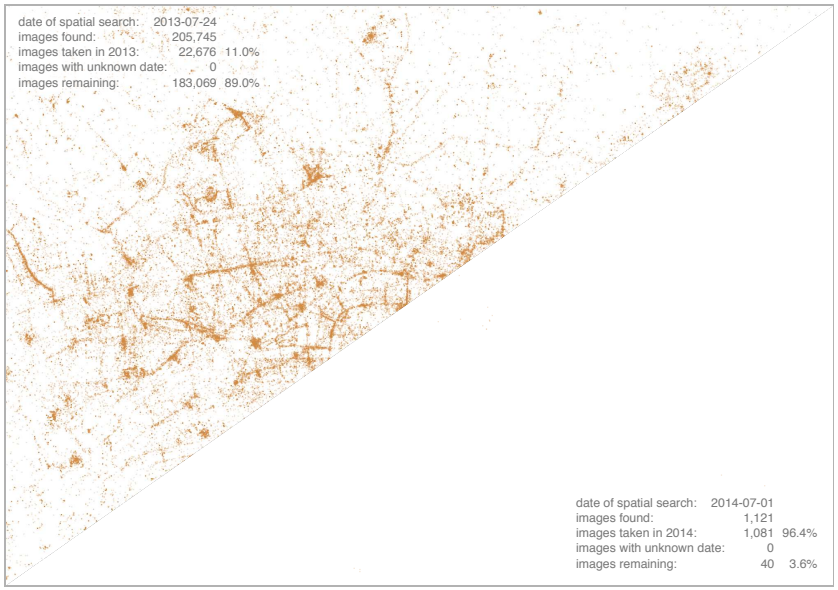


Figure 4.18: The latest versions of initial photographic distributions. All datasets are pre-filtered by date of photographing – only images reported as taken before January 1, 2014 are shown.

4.3 Distribution analysis

The reliability of a distribution of 'votes', which forms attractiveness scores for street segments, depends on many factors. Even if all images are daytime outdoor photographs taken by pedestrians (as it is determined in a set of requirements on page 38), a routing algorithm that is based on these data can be misled due to bias of different nature and thus not give preference to the most attractive streets in variety of situations. Some areas may potentially become over-represented because of the inaccuracies in data, while other may get redundant 'votes' due to peculiarities in behaviour of contributors.

This section considers cached photographic datasets as a whole and includes the results of investigation of their structures. As suggested by the methodology, this part of analysis solely deals with coordinates of images, ignoring their content and attached metadata. Only the following four attributes are taken into account: image id, location (spatial coordinates), time of photographing (temporal coordinate) and authorship (photographer's name and id).

4.3.1 User activity analysis

The idea of estimating street attractiveness using spatial densities of crowd-sourced geotagged photographs implies relying on subjective views of individuals, which makes it important to understand who the data are formed by. Certain common patterns in photographers' behaviour may have negative impact on the derived attractiveness scores.

In a *model photographic collection* (page 20) the 'votes' are received from equally engaged users (requirement 4), and this guarantees that differences in personal favours are averaged and smoothed. Thus, if few photographers appear to have extraordinary preferences and choose places that can be hardly considered as attractive by others, their contributions are compensated by those from 'average' users and do not cause radical changes the results of route generation. Previous research (e.g. Purves, Edwardes and Wood 2011) demonstrates that in real crowd-sourced collections of photographs there exists a 'participation inequality' effect (Nielsen 2006), meaning that a small proportion of contributors provide most of the data. This effect was confirmed in cached photographic distributions and is demonstrated in Figure 4.19.

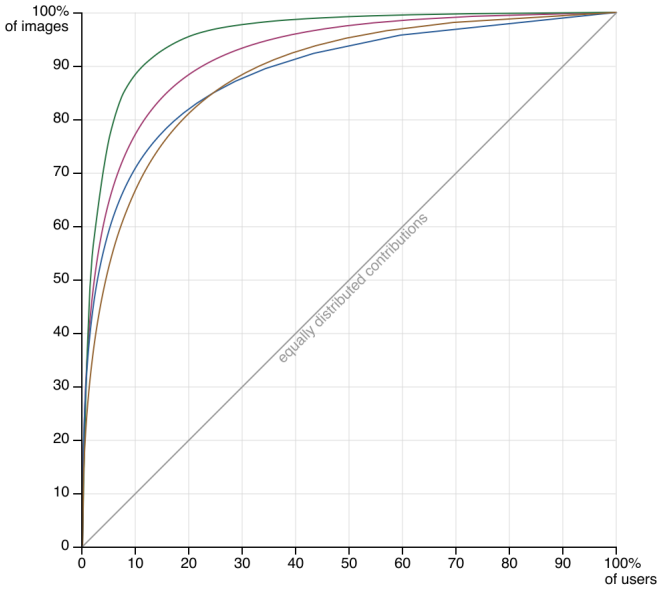


Figure 4.19: Inequalities in contributions by Flickr, Geograph, Panoramio and Picasa users.

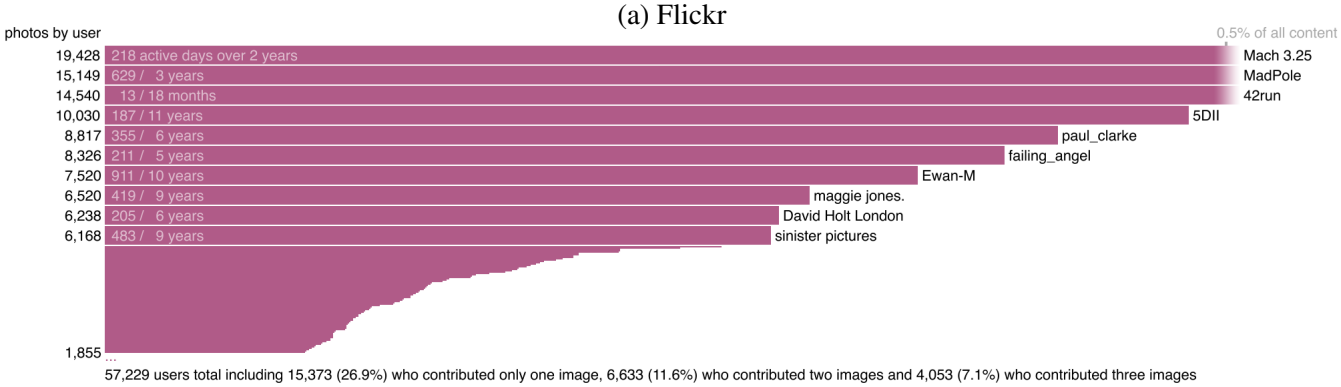
Participation inequality in Central London was found the highest in Geograph, where 10% of users contributed nearly 90% of all images. Photographers in Picasa (rejected earlier) and Panoramio were engaged the most uniformly, but still remained far from what could be expected from a *model photographic collection* (such a dataset would look like a diagonal line on the above chart).

The fact that a small proportion of users generated vast majority of potential 'votes' suggested to examine individual contributions of the most active users and propose generalisable rules of reducing their impact on the values of street attractiveness scores. Figure 4.20 on the facing page exposes the volume and other properties of such contributions.

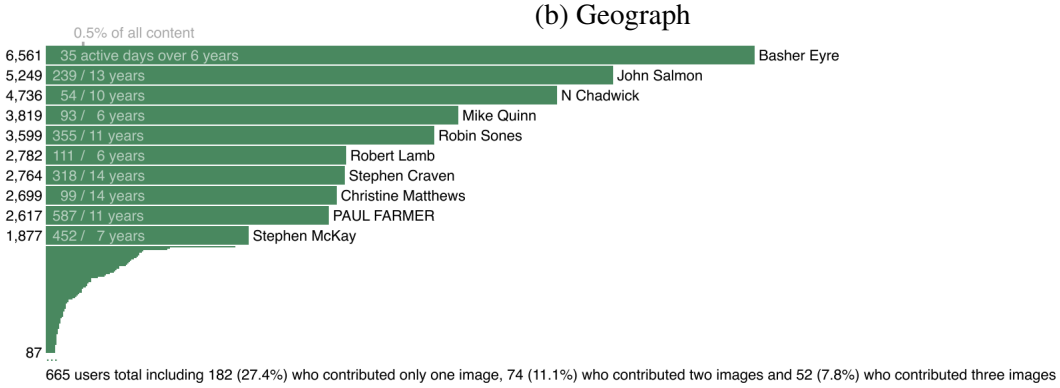
As it is seen from the bar charts, the most active Flickr photographers share the highest numbers of images among the users of all four sources of data. However, the proportion of their images in the overall collection is the smallest (note the position of '0.5% of all content' mark), and the total number of users is the highest. These two facts make bias by individual preferences in Flickr significantly less likely than in Panoramio and Geograph, the latter of which is formed only by 665 individuals.

Exclusion of the most active users from a potential source of street attractiveness scores may seem to be the most straightforward solution for bias reduction, but this method is difficult

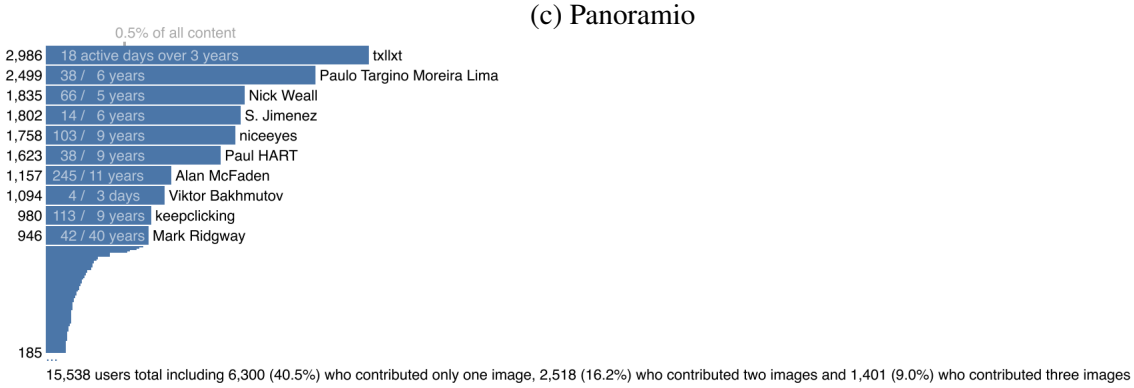
(a) Flickr



(b) Geograph



(c) Panoramio



(d) Picasa

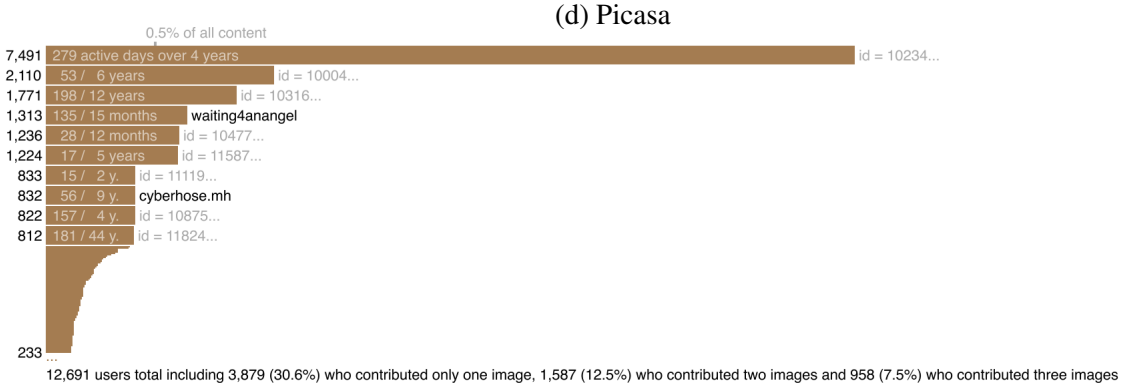


Figure 4.20: Individual contributions from 100 most active users in each photographic dataset.

to optimise. Shapes of 'tails', which are formed by top 100 active contributors in all cached datasets (also Figure 4.20), show that there are no vivid gaps between the users who can be considered as outliers and those that are active, yet not in the very top of the list. Thus, setting a threshold of, let's say, a thousand of images or 0.5% of all content as a discriminating factor for filtering cannot work well as a robust solution. This number would need to depend on the size of a chosen area, the origin of data and some other factors. Besides, the fact that a person is actively sharing photographs of locations of their interest, does not reduce the value of their preferences to the routing algorithm – the question is how to reliably calculate the submitted 'votes' for street attractiveness.

The answer to this question was proposed after examining spatial distributions of photographs left by individuals (examples can be found in Figure 4.21 on the next page). It was observed that users who take and share large numbers of pictures tend to adhere the following three common patterns:

Mass photographing along particular walkways (e.g. Flickr user *March 3.25*, Geograph user *N Chadwick*, Panoramio user *S. Jimenez*). Subsets of photographs are closely located next to each other, forming clearly distinguishable paths. Few images have exactly the same coordinates. In many cases temporal coordinates are not distant, suggesting that the photographs were taken continiously during one or several walks.

Photo scattering. (e.g. Flickr user *Ewan-M*, Geograph user *Robin Sones*, Panoramio user *keepclicking*). Spatial and temporal coordinates of images are less organised and do not produce any well-formed shapes. Photographs cover relatively large areas and are also scattered over time (the proportion between the days with activity and days of presence is usually larger than in the first scenario). This suggest that the 'votes' may be left after more walks than in the first case, but their number per walk is much less (of course, if the walks did take place, and the images were not taken during events, indoors or else).

Location averaging. (e.g. Flickr user *42run*, Geograph user *John Salmon*, Picasa user *cyberhose.mh*). Significant numbers of photographs share identical spatial coordinates, even when the photographs have distant timestamps. This suggests that users who follow such a pattern pay less attention to accuracy of geotags they define, and it is probable that the majority of the 'votes' are not related to the areas where they are located.

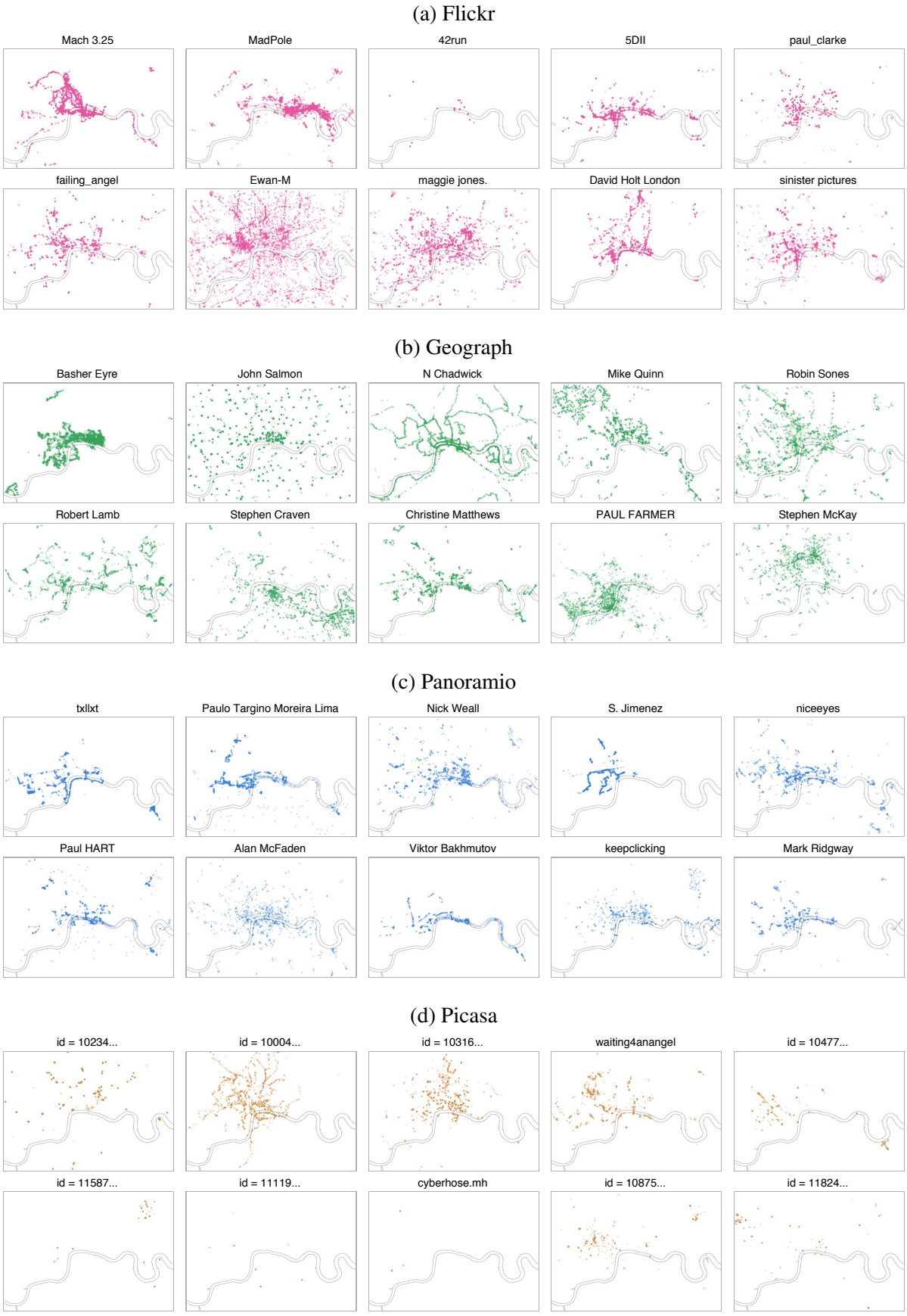


Figure 4.21: Locations of photographs from ten most active users of each photo-sharing service in Central London.

According to requirements 3 and 5 on page 38, which define the properties of a *model photographic collection*, filtering is required in all three cases. As each photographer should take maximum one picture at any place (requirement 5), the number of images in the first scenario should be reduced, so that in the resulting spatial distribution the distance between individual photographs is not less than a certain value, and the same street segment does not get unreasonably many 'votes' from one user. The same applies to the second type of behaviour if the density of scattered images is too high (assuming that all photographs are valid from the perspective of their content). Photographs with averaged locations, however, cannot be considered as adequate 'votes' as they violate requirement 3 (attached georeferences should be valid), so such contributions should be excluded from the data.

The above patterns are often combined in individual contributions (e.g. see Flickr user *sinister pictures* or Panoramio user *Mark Ridgway*), which makes it impossible to accurately group users by the type of their behaviour in order to design and test behaviour-specific filtering methods. Besides, choosing what photographs need to be kept in the first and the second scenarios may be a challenging task, because a relatively small distance between the remaining 'votes' may still bias the attractiveness scores for long street segments, and its increasing may result unwanted loss of some of the 'votes'.

As an alternative solution, it was decided to retain photographs at all locations except those that are quite likely to be inaccurate (scenario 3), but let the density-to-score mapping function M (page 21) consider the number of users rather than the number of photographs as what forms the street attractiveness score. This approach made requirement 5 (no more than one 'vote' from a user at a place) satisfiable in all situations, but did not introduce any complex filtering procedures. Thus, existing inequalities in contributions can be significantly suppressed – even the most active user in the entire photographic dataset can add at most one 'vote' to each road segment, which in the worst case scenario is equivalent to adding insignificant amount of noise to the attractiveness scores.

The second observation that could be made when profiling photographers was that a lot of individuals in all four datasets submitted only one image. The proportion of this number to the overall population of contributors was the highest in Panoramio data, being equal to 40.5% (Figure 4.20 on page 139). There was a danger that these could be users who simply 'try out'

a photo-sharing service (Purves, Edwardes and Wood 2011), thus submitting high proportions of irrelevant images that could be hardly considered as 'votes'. Manual browsing of these data showed that such cases were rather rare, and filtering of unbundled photographs could not increase the reliability of the derived attractiveness scores. On the other hand, reduction of the population of users would make each individual contribution more significant and thus increase noise, so filtering could even have negative consequences.

Another short study of photographic data at the level of individual contributions was held during the research. This analysis was inspired by Fischer (2010), where Flickr images that were geotagged in over a hundred cities worldwide were placed on a map and coloured by user category: *tourist, local* and *unknown* (Figure 4.22). These categories were assigned to photographers based on the durations of their presence in the exposed area. The visualizations have demonstrated that people who are active for smaller periods of time tend have different preferences from those who contribute on more regular basis. This discovery prompted to look at this phenomenon in more detail.

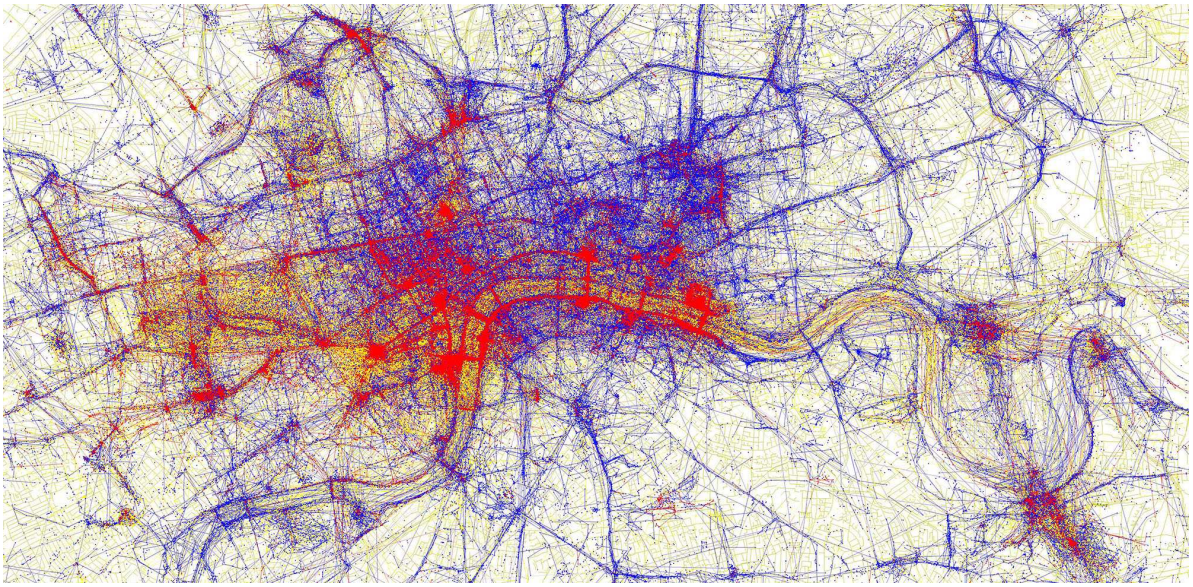


Figure 4.22: 'Locals and Tourists in London' – a visualization based on Flickr and OpenStreetMap data. *Source: Fischer (2010), CC-BY-SA 2.0.* Photographs are represented with dots and are red if they belong to a 'tourist', blue if contributed by a 'local' and yellow if user category has not been established. Line segments link images that are taken by the same user in order from the earliest to the most recent time coordinate.

Figure 4.23 shows the relationship between the sizes of individual contributions, period of user's presence (time interval between their first and the last photograph) and the numbers of days of their activity (days with at least one photograph) in all four considered datasets. Such representation highlights similarities in their structure and also reveals a few differences. In all cases there exists a cluster of users who were engaged with photographing for no longer than one or two weeks and did not share large numbers of images. This group, located in the bottom-left corner of each chart, is the smallest for Geograph – it mostly consists of users who were involved in photographing only for one day according to the attached time coordinates. Flickr appears to contain the richest variety of users, but this can be explained by the higher popularity of this service as such.

Following Eric Fischer's idea of distinguishing between 'locals' and 'tourists', it was decided to look at users with different footprints of activity and presence and thus confirm or deny variations in behaviour that were observed earlier. With two derived time coordinates instead of one the following three nominal categories were proposed: *casual users* (less than 14 days

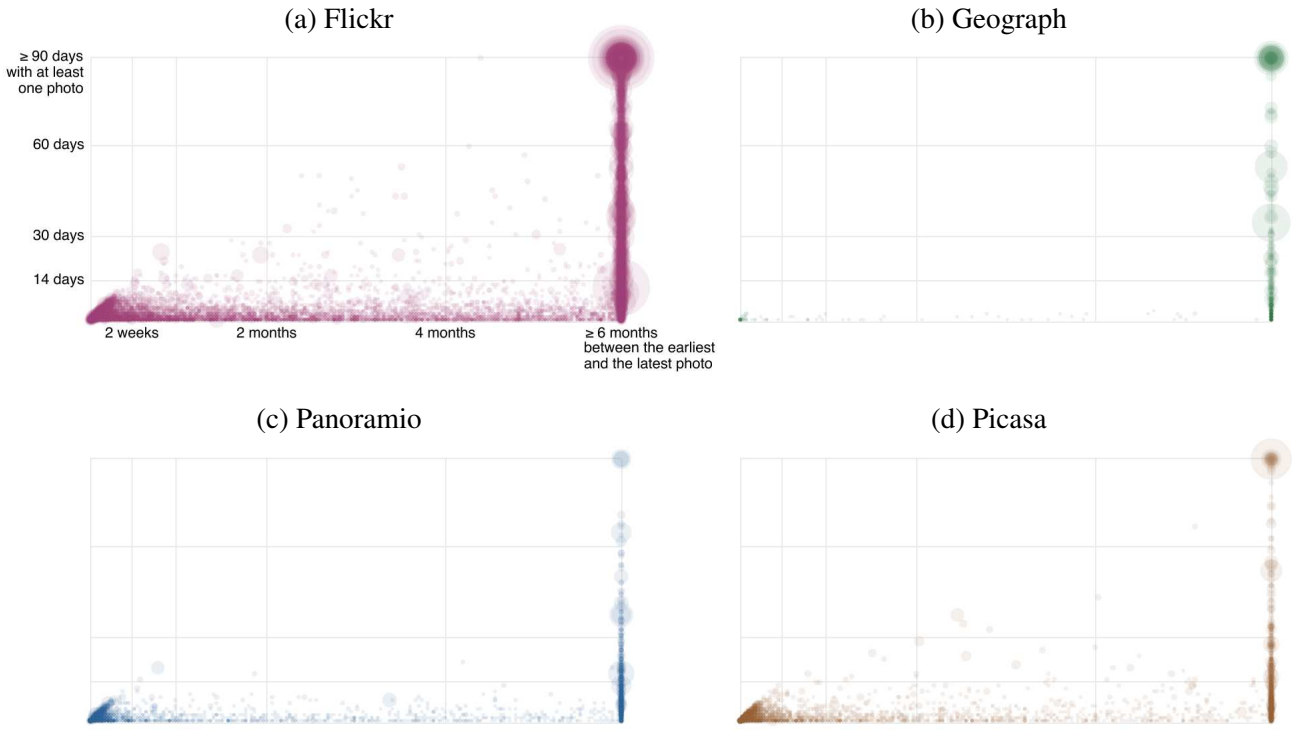


Figure 4.23: Individual contributions broken by users' period of activity and presence. Every circle represents one user and is sized according to the total numbers of photographs taken.

Figure 4.24: Proposed derived categories for users of photo-sharing services.

of activity and presence), *returning users* (not more than 14 days of activity, but more than two months of presence) and *regular users* (more than 30 days of activity and two months of presence) as shown in Figure 4.24. Expectancy maps (Figure 4.25 on the following page) helped see the differences between these categories in all four datasets.

As in the visualization on page 143, the contributions that belonged to *casual users* (the equivalent of 'tourists') appeared more often than expected in the most central parts of London such as City and Westminster – the tendency could be observed in all four sources of data and was particularly strong in Flickr. Casual photographers were forming the largest batch (from 49.9% in Geograph up to 79.6% in Panoramio), which, however, did not necessary contain the highest numbers of images. Groups that united *regular users* (the equivalent of 'locals') were opposite: the distributions of contributions were shifted towards residential areas, and the numbers of users were rather small. The total number of photographs were not proportional and reached from 11.9% in Panoramio to 74.8% in Geograph. The behaviour of *returning users* differed from source to source, being similar to *casual users* in Flickr and Geograph and less apparent in Panoramio and Picasa.

Thus, filtering users by activity and presence seemed to have a potential to influence the distribution of road attractiveness scores. Dealing only with *casual users* could make leisure walks appear more often in busy parts of the city, while working barely with *regular users* could shift them towards quite residential areas. This peculiarity may have good chance to find application in real photo-based routing systems. However, the side-effect of such filtering can be violation of requirement 4, which states that a collection of 'votes' should be formed by a reasonable number of photographers. Among the datasets this project is dealing with, this

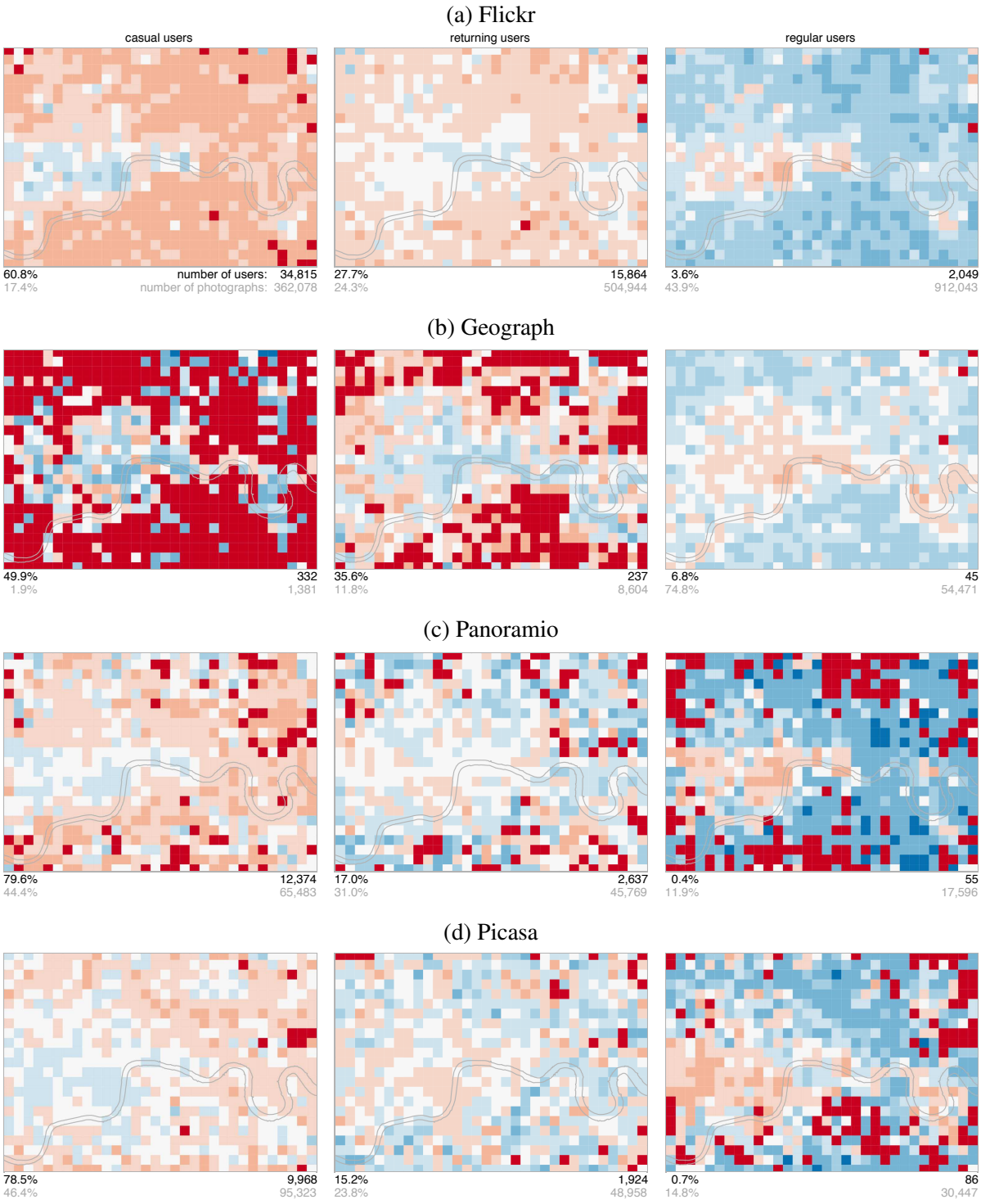


Figure 4.25: Local user number expectancies by their category in comparison to the overall activity. Red squares denote areas where more contributions from the users of a given category were expected than were actually found. Blue squares correspond to areas where the proportion of users of a given category is higher than it is expected to be. *It can be observed that contributions from casual users are more likely to be found in the city centre while regular users tend to be relatively more active in other neighbourhoods.* The size of a square is 250×250 meters. Picasa map may be inaccurate due to incompleteness of the data.

situation, perhaps, would not happen only in Flickr, where there are several thousand users in all three nominal categories.

The following sections consider data from all photographers without dividing them.

4.3.2 Analysis of temporal coordinate

Although function M, which converts a distribution of geotagged images into attractiveness scores, does not consider time of photographing as such, this attribute can play an important role in making available data less distant from a *model photographic collection*. Being attached to most of the images, temporal coordinate provides an opportunity to check potential sources of street attractiveness scores against some of the requirements and apply filtering of the 'votes' if there is a need. This subsection examines patterns in values of time of photographing for cached metadata and makes filtering suggestions solely based on this attribute (i.e. without considering spatial coordinates).

Two key views, presented in Figures 4.26 and 4.27, reveal the most general trends in the data.

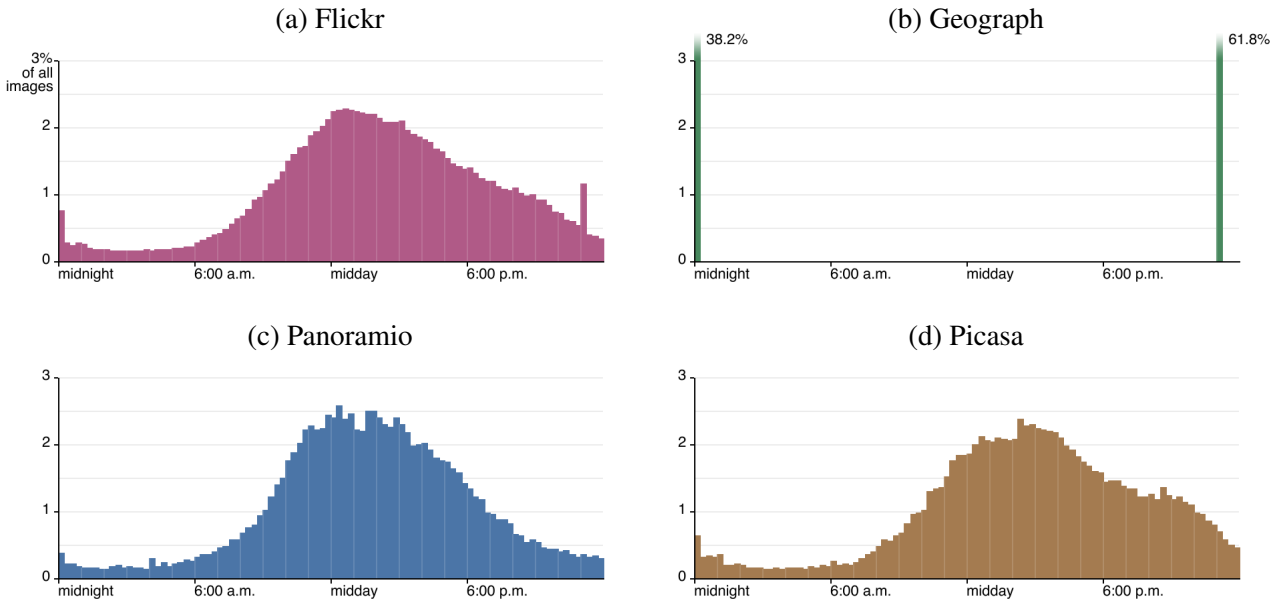


Figure 4.26: Distributions of all photographic records by time of day. Time zone is GMT+00 (Greenwich Mean Time).

The first view (Figure 4.26) shows how the images are distributed around the clock. Every hour is split into four intervals 15 minutes each, and the popularity of all 96 intervals is then counted. The most noticeable artefact in the resulting distributions is the existence of only two bars in Geograph data – one at midnight and another at 11 p.m. Such 'abnormality' can be easily explained by the fact that this source of image data only stores day of photographing, ignoring exact time. The value becomes equal to 23:00:00 in about 60% of cases, because time shifts by one hour in summer season for daylight saving. Thus, more than a half of the photographs become tagged as taken one day earlier if time zone correction is not applied.

Flickr, Panoramio and Picasa share similar time profiles. The peak of activity in these datasets takes place after midday and is followed by a slow decline that stops around midnight. Flickr and Picasa distributions are sightly negatively skewed, which indicates more activity during the evenings. Unlike Panoramio that declares itself as a 'site for exploring places through photography', these two image-sharing services can be used for any legal purpose, so pictures from parties and other social activities are more likely to appear there. Cases of absence of time in the temporal coordinate (such as in Geograph) also exist in three other datasets – peaks with distinguishable magnitudes can be observed at midnight in Flickr, Panoramio and Picasa, and also at 11 p.m in Flickr. Manual exploration of such images confirmed that few of them could be taken at this time.

The ability to filter photographs taken at 00:00:00 and 23:00:00 together with a fact that the observed profiles of daily activity have explanation, do not guarantee the validity of temporal coordinate. This attribute is either defined by a camera or manually added by a user at the moment of photo sharing, and in both cases the mistakes may occur. Even when a photographer has no intention to submit incorrect data, and built-in camera clock has been configured since the last reset, time can be shifted by any number of hours due to an inaccurately defined time zone. Given high quantities of *casual users* in all sources of data, it is not fair to assume that such cases may be rare – a proportion of non-regular contributors can be tourists who come from different parts of the world and have an unknown time zone setting with unknown probability. Estimating the impact and the nature of possible time shifts is hardly possible – there is no 'ground truth' in the contents of the photographs or other metadata that could help precisely asses attached temporal coordinate. The existence of shifted time coordinates, however, can

be observed in Flickr, Panoramio and Picasa charts as a non-changing activity level between midnight and approximately 6 a.m. If time attribute was always precise, one could expect the heights of the bars within this interval to be uneven as human affairs tend to go down towards the end of night.

Summarising the above, it can be concluded that although the exact time in temporal coordinate does have some degree of reliability in three out of four data sources, it can hardly be useful in filtering. Exclusion of nighttime photographs (requirement 8) based on moments of sunrise and sunset will keep the majority of entries, but not remove the noise that is brought in by possibly incorrect time zone settings or for other reasons. Besides, this approach to filtering cannot guarantee that remaining daytime photographs are taken outdoors (requirement 7).

Distributions of temporal coordinates at a different scale is shown in Figure 4.27 on the next page. Here the photographs are arranged into groups that represent 5114 days starting from January 1st, 2000 to December 31st, 2013. The second limit corresponds to the upper time boundary, defined for the latest version of the photographic datasets in Subsection 4.2.1. The earliest date has been picked to incorporate most of the cached data – only few hundred images from several dozens of users are reported to be taken earlier. The view is supplemented with numbers of unique users who form sets of photographs in each group. Such representation led to the following observations:

Most of the cached photographs are reported as taken after the launch of each photo-sharing service, and their number tends to increase over time. There is a soft decline in the ‘popularity’ of Panoramio and Flickr in 2013, but this effect can be at least partially caused by gaps between date of photographing and date of sharing, demonstrated in Figure 4.11 on page 127.

Arrays of both photographs and users have relatively high amount of deviation and also contain outliers. Extreme volumes of images in a set of days can be explained by contributions from the most active photographers – peaks in the upper charts do not always match local maximums in numbers of users below. However, spikes in lower charts may be signs of changes in the conditions of the environment such as events.

Periodic inclines and declines can be observed in some of the series. These can be signs of seasonality in the distributions.

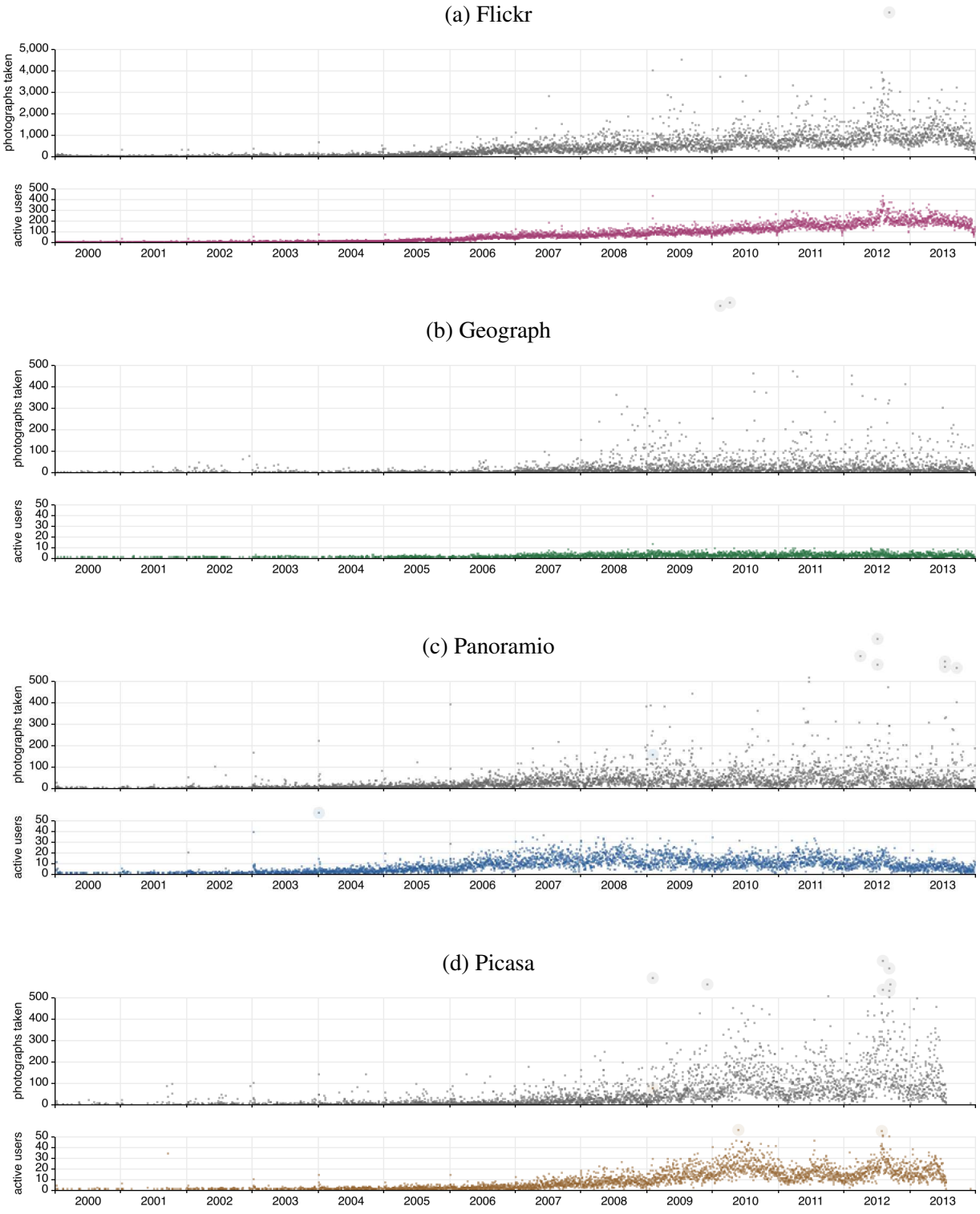


Figure 4.27: Temporal distribution of the daily photographers' activity.

In response to the first observation, it was suggested to introduce a lower boundary for time coordinate in addition to the upper one, which was determined due to a gap between dates of image taking and sharing. Exclusion of photographs from early years did not significantly reduce the sizes of the datasets (so does not make it less suitable), but helped protect the routing algorithm from potentially irrelevant 'votes'. Manual exploration of early photographs suggested that many of them were simply inaccurately referenced, however even if all data were 100% correct, exclusion would be still reasonable. As landscapes of cities undergo the changes over years, relying on old 'votes' for attractiveness may distort the desired outcome.

In this project it was decided to use January 1st, 2008 as a lower time boundary, resulting six full years worth of data in the latest cached collections of photographs. This range, however, may not be used as a fixed recommendation – its size can depend on a variety of factors such as the volume of available photographic data, level of 'conservatism' in the landscape of the chosen region, necessity to apply different other kinds of filtering, etc.

Detailed exploration of outliers in temporal distributions of images confirmed that many of them were caused by individuals who simply took many photographs on an arbitrary day. These peaks could not bias street attractiveness scores, as it was decided earlier that M function does not count every photograph in the surroundings and considers numbers of neighbouring active users as 'votes' instead.

Outliers in numbers of users were classified into two categories. The first can be easily found in the charts as spikes at the beginning of calendar years, especially the earlier ones. These groups are formed by the images with 'rounded' time coordinate, e.g. 2003-01-01 00:00:00. Such photographs are believed to be tagged manually with some human interfaces that allowed users to define only the year of photographing and were filling the rest with the defaults. Such interfaces are quite likely to be discontinued later, because rounded time coordinates appear less often in more recent years.

Peaks of the second category can be explained by the changes in the context of photographing such as global and local events. Unlike with cases of vast numbers of pictures from single individuals, sudden changes in activity of the whole community does have a potential to decrease the reliability attractiveness scores. If an outlier is caused by a temporal phenomena that at-

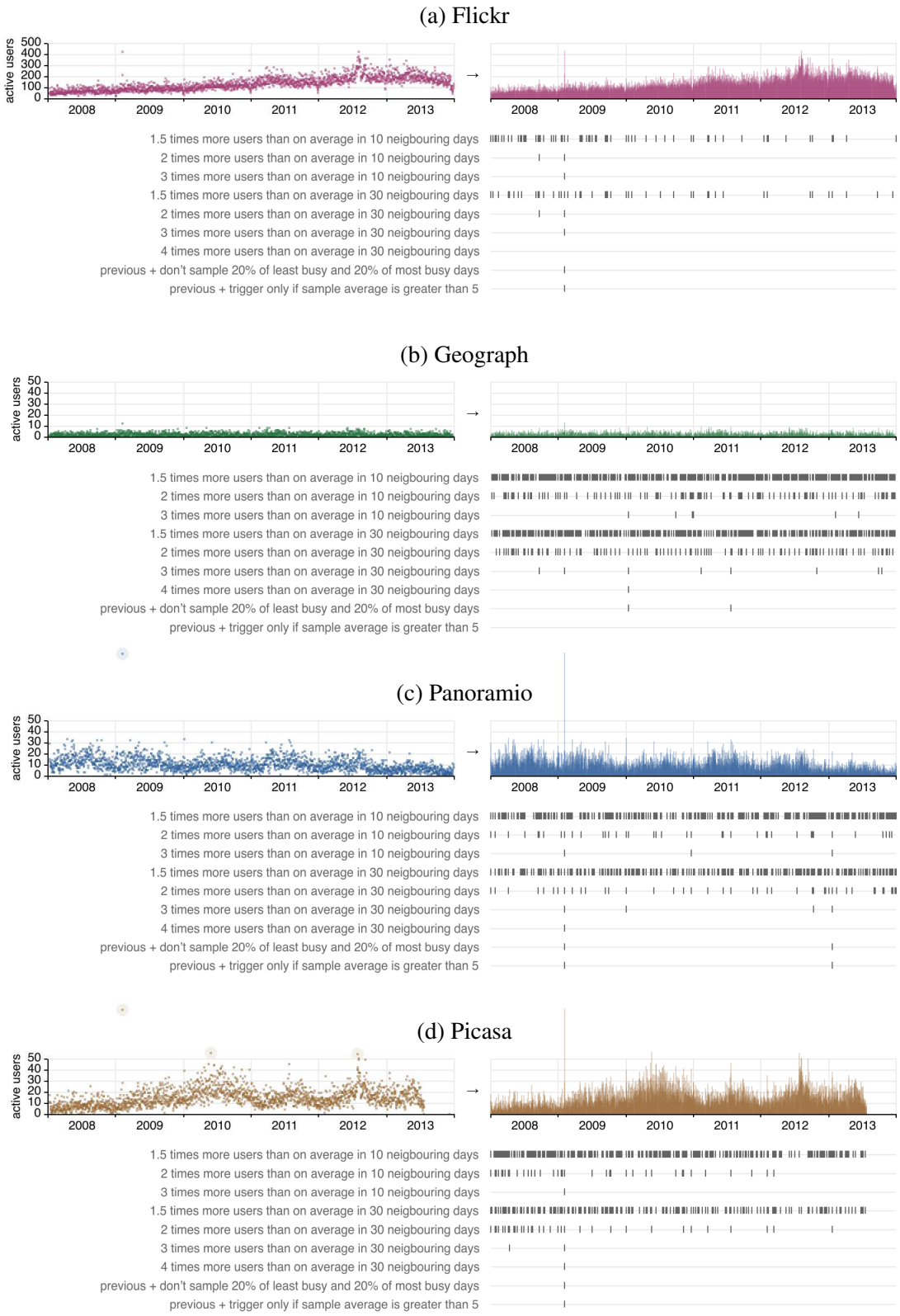


Figure 4.28: Global event detection using rules with different parameters.

tracts significant amount of photographers' attention, surrounding street segments receive extra 'votes', which should be considered as irrelevant by design (requirement 10). Importance of filtering in such situations can be demonstrated on a fictional example – an asphalted square in an industrial zone, occasionally used as a site for open-air concerts. If the contributions that are made at this place during only a few days over several years are treated the same way as more evenly distributed pictures in a neighbouring park, a routing system will consider both locations equally attractive for leisure walks. Obviously, such cases should be avoided.

Temporal anomalies (or events) can affect vast geographical areas, which suggests classifying them into *local* and *global*. Figure 4.28 on the preceding page highlights global peaks in temporal distributions of user activity and demonstrates how different rules could be applied for excluding days with extremely abnormal numbers of users. As it is seen from the graphics, utilisation of a temporal coordinate for filtering on its own has a rather narrow application and can be only used for detection of significant global events. In the given datasets the only case that could be stably detected was a heavy snowfall, which took place in London on February 2nd, 2009 (BBC 2009*a*; Gillan 2009). Detection of less impacting large-scale events such as protests, marathons, fireworks, etc. appeared to be impossible because of a relatively high variance in typical levels of user activity. Long-lasting events such as the Summer Olympic Games in August 2012 were found undetectable as well, neither with a fixed static threshold for numbers of users, nor with adjustable sliding windows. Such limitations in potential use of the time coordinate as an instrument for filtering together with a need to process occasional local events necessitated an alternative approach to data cleaning in these situations. The method is introduced later in Subsection 4.3.4 on page 169.

Seasonal irregularities in photographic datasets were previously raised in some research projects (e.g. Andrienko et al. 2009; Alivand and Hochmair 2013; Yamasaki, Gallagher and Chen 2013) and could be also be observed in temporal distributions of gathered data. A new visual representation was designed to expose this effect (Figure 4.29 on the next page). With both numbers of photographs and users grouped by years and months, it describes each group with five statistical measures and shows aggregated summaries for six most recent years. The same multidimensional layout was used to study irregularities by days of week (Figure 4.30 on page 155).

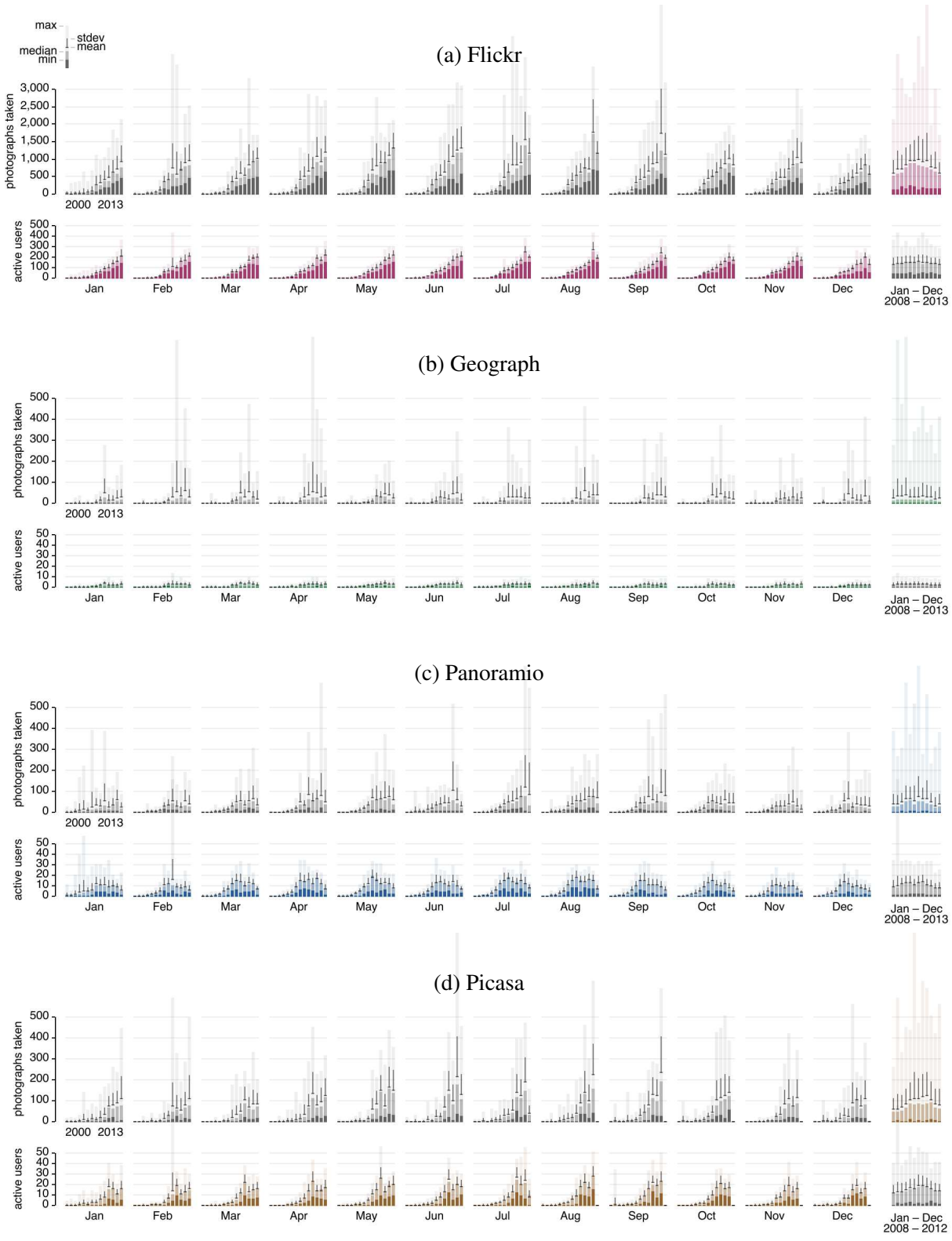


Figure 4.29: Temporal distribution of photographers' activity aggregated by years and months.

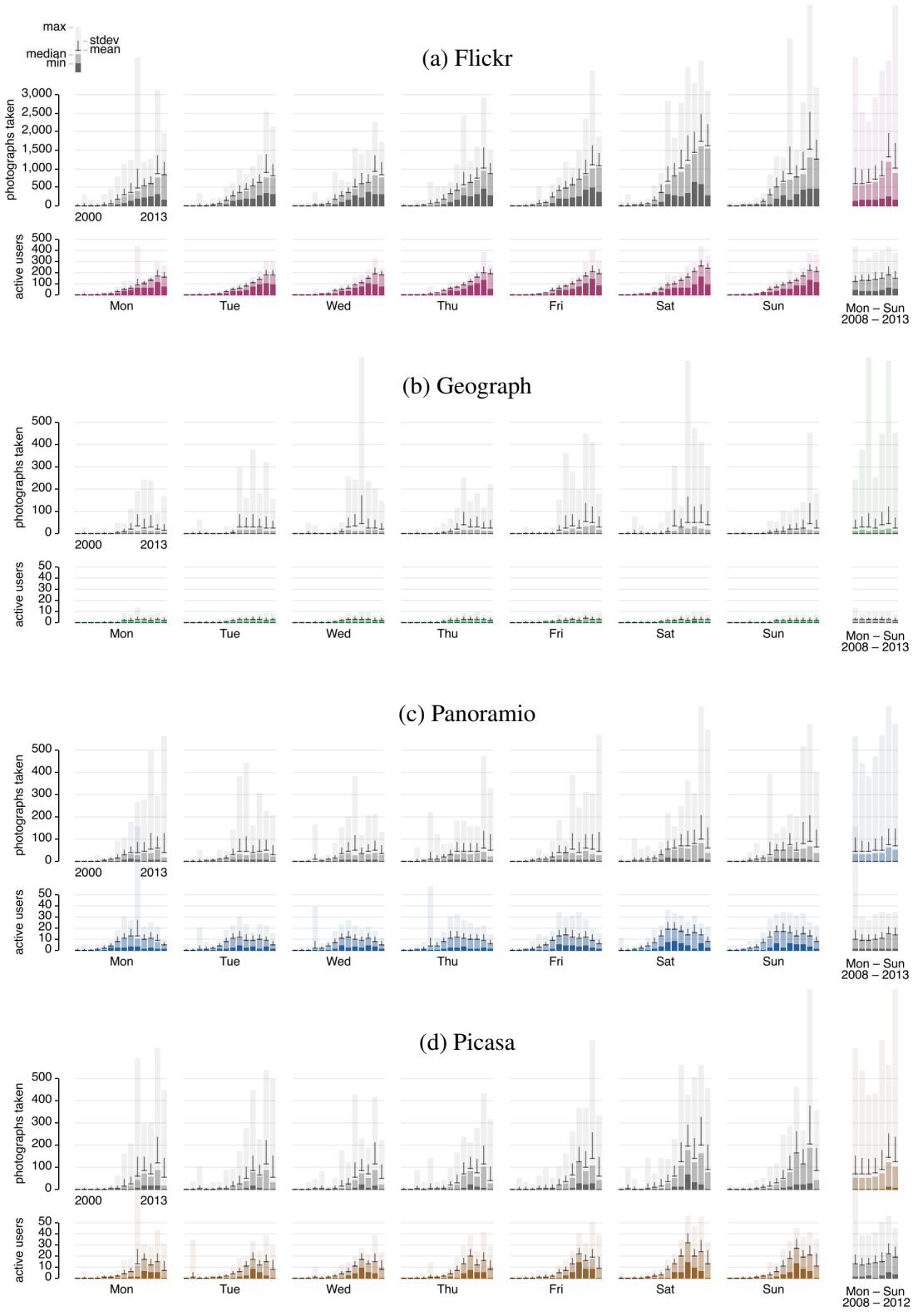


Figure 4.30: Temporal distribution of photographers' activity aggregated by years and days of week.

As it can be confirmed from the first set of charts (page 154), seasonality does take place in London, but its amplitude is not overwhelming. The biggest difference between levels of user activity in summer and winter can be observed in Panoramio data with about 50% of active users in June more than in December, and for Flickr the variation is about 30%. Popularity of Geograph is rather constant around the year. Taking into account that weather in London does not radically change over the year due to its oceanic climate (Met Office 2014) as well as the fact that tourist flows are also relatively evenly spread (ONS 2014), it can be speculated that similar charts may show higher variations of values in other places. Extremely cold winters, hot summers or rain seasons may potentially cause significant dissimilarities in activity of photographers in different months, thus making overall attractiveness scores less reliable. Filtering of images by time of year is technically possible and can be recommended for cities with continental or tropical climate. Narrowing data to a few months, however, may not be suitable for relatively small photographic collections such as Geograph or Panoramio, because extensive filtering may significantly increase relative amount of noise and bias from personal preferences, thus requirement 4 will be no longer met.

Aggregation of photographs and users by day of week (Figure 4.30 on the previous page) did not reveal any suspicious patterns. It was observed that user activity over the weekends (Saturdays in particular) was slightly higher than during the working days in all datasets. This fact could not be interpreted as abnormal behaviour, which would require further investigation and cleaning. Nevertheless, filtering by day of week could be useful in some rare situations.

Exploration of temporal coordinates in photographic data was an important step towards understanding selected sources of images as potential base for street attractiveness scores. London, being a city with relatively stable tourist flows and mild climatic differences between seasons can be probably described as a case where not much filtering by time of photographing is necessary – only pictures reported as taken before 2008 and on February 2nd, 2009 were removed. Automated exclusion of old images as well as from days with extreme user activity may be potentially sufficient for data processing in many situations, but it can be recommended to check distributions from other regions against the discussed causes of bias to ensure in their reliability. In any case, because filtering by time of photographing is not able to help with handling small-scale events, another approach to filtering may be necessary.

4.3.3 Analysis of spatial coordinates

Sufficient accuracy of spatial coordinates (or a geotag) is a crucial requirement to a collection of crowd-sourced photographs that may be potentially considered a reliable source of street attractiveness scores. The more images appear to be distant from a location where a photograph was actually taken, the more noisy attractiveness scores become. Any systematic inaccuracies can result bias, making particular street segments more attractive than they are due to presence of non-randomly dislocated irrelevant 'votes' in their neighbourhood.

The theoretical level of accuracy of a geotag depends on a method that has been applied to create it – this is done either automatically by a camera or manually by an author of an image with use of a human-computer interface. Spatial precision in the first case is conditioned by technical limitations of a GPS receiver or a combined method of location detection (e.g as in Apple 2014), giving ambiguity from a few meters in open spaces up to 40–50 meters in 'urban canyons' (areas with skyscrapers) where signal is poorly available and is reflected from surrounded buildings (U.S. Air Force 2014). Such accuracy, being 10-20 meters on average, can be considered as sufficient for the chosen purpose. The second method of geotagging gives less predictable results and depends on the photographer's intention and experience as well as a human-computer interface that is being used. Thus, a part of manually tagged images may be characterised with very high precision of spatial coordinates, while an unknown number of images can be significantly misplaced.

The method of geotagging cannot be unambiguously identified in most of the cases. When this process is done automatically by a camera, coordinates are saved as EXIF tags, which are then read by a photo-sharing website and used for referencing. Desktop photo-editing software, however, adds the same tags to EXIF when a picture is placed on a map manually by a human. Only when a geotag has been added after an upload, absence of latitude and longitude in EXIF can give a clue about their origin, assuming that these attributes can be always read from EXIF when exist. Such peculiarity of the process of georeferening makes it impractical to distinguish between automatically and manually tagged photographs and thus excludes an opportunity to rely on the accuracy of spatial coordinates differently depending on their origin.

The question of spatial accuracy of crowd-sourced photographs was raised in a number of empirical studies such as Hochmair (2010), Zielstra and Hochmair (2013) and Hauff (2013),

where locations for small random samples of Flickr and Panoramio images were checked against 'ground truth' proposed by the authors. These studies conclude that although median displacement of all checked photographs is sufficiently small (within 50 m), some images (especially those from Flickr) are tagged several hundred meters away from the photographers' actual position, as it can be known from the captured perspective of the scenery. According to Hauff (2013), accuracy of geotags in more popular locations (such as city centres) is generally higher than in rural areas.

In this research it was decided to study spatial accuracy of cached data records by means of visual analytics and thus detect photographs that could be geotagged with bias (violation of requirement 3 on page 38). With visual analytics it was possible to find common misplacement patterns and suggest filtering rules based on some features of spatial coordinates or mutual positions of the images. The analysis was mostly done by means of custom software, which was described in Subsection 3.3.2. Granularity (maximum theoretical accuracy) of all cached geodata was equal to 10^{-6} for both latitude and longitude, matching the highest precision of coordinates in photo service APIs. 0.000001 of a degree in WGS84 at the 51st latitude (i.e. near London) is 7 cm from East to West and 11 cm from North to South.

The most straightforward inconsistency that was found in cached photographic datasets during exploratory analysis was an anomaly at WGS84 Prime meridian (Figure 4.31). It was only observed in Flickr, but was persistent in all versions of this dataset. The nature of such anomaly

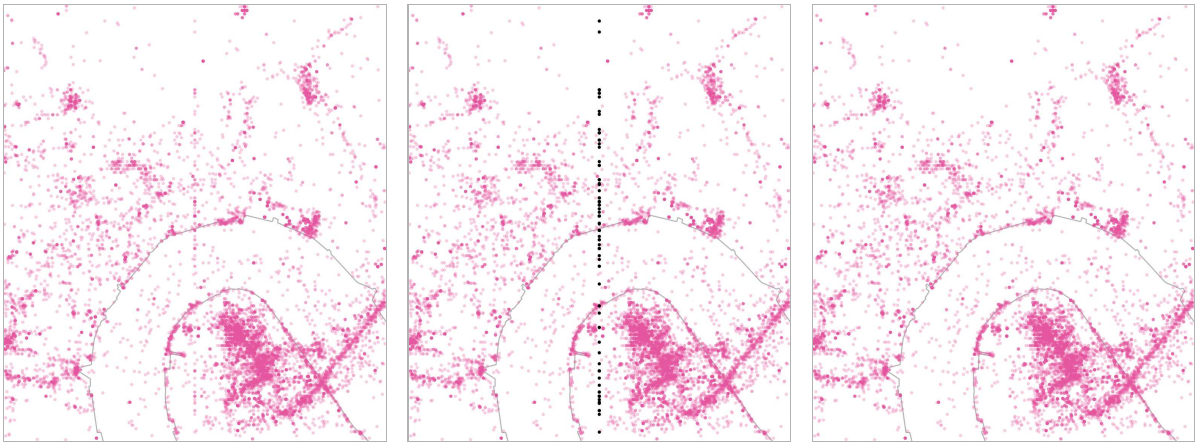


Figure 4.31: Locations of Flickr images near Prime meridian. *Left:* original distribution; *middle:* 265 images by 67 photographers to remove (some photographs are outside of the featured area); *right:* data after filtering.

suggested that it could be caused by some accidental errors in data, which made longitude equal to zero. Manual exploration of images at Prime meridian showed that although most of them were taken in London, few were made in areas that were specified. In all observed cases longitude was equal to zero at the origin (i.e. on Flickr website), which meant that the anomaly was not caused by any error in the API or in local software.

Simple exclusion of all images with WGS84 longitude equal to $0° \, 0' \, 0''$ was applied to reduce bias in areas near Prime meridian. Although such filtering could also affect several accurately placed photographs, their removal would not have any significant negative influence, as Figure 4.31 visually demonstrates. In fact, because spatial coordinates in all chosen datasets except Geograph have precision of up to 10^{-6} of a degree, any 'bin' on longitude axis has width of only seven centimeters and thus contains only a very small fraction of geotagged images that describe the neighbourhoods.

Another anomaly that could be discovered during interactive visual exploration of all four photographic datasets was presence of hotspots – points with unexpectedly high numbers of records (an example is shown in Figure 4.32). Their existence would not be possible if all photographs were geotagged with a normally distributed displacement and identical precision of the coordinates, so the case required careful investigation. As it was pointed in a study of user behaviour in Subsection 4.3.1, hotspots could be a result of mass geotagging, but that observation was not sufficient to make conclusions about all of them – some could be formed

Figure 4.32: One of the hotspots detected with Photo Distribution Viewer. The view on the right has smaller opacity of circles and an added coordinate variance.

by rounding of spatial coordinates or other reasons. Knowing the natures of the hotspots was crucial for making conclusions about which of them introduce bias in spatial distribution of images and should be therefore excluded.

It was found that some hotspots as well as individual images were arranged in a form of grids, as demonstrated in Figure 4.39 on page 171. The resolutions of these grids and their clearness were different, which suggested that they could have various origins. During the investigation of the observed grids, all could be explained by the choice of the original notation for spatial coordinates.

In Geograph all images are always georeferenced using Ordinance Survey National Grid, which has a varying edge of one, ten or a hundred meters. After the coordinates are converted to WGS84 with granularity of 10^{-6} degree (0.07 and 0.11 m in London for longitude and latitude, respectively), grid cells remain visible.

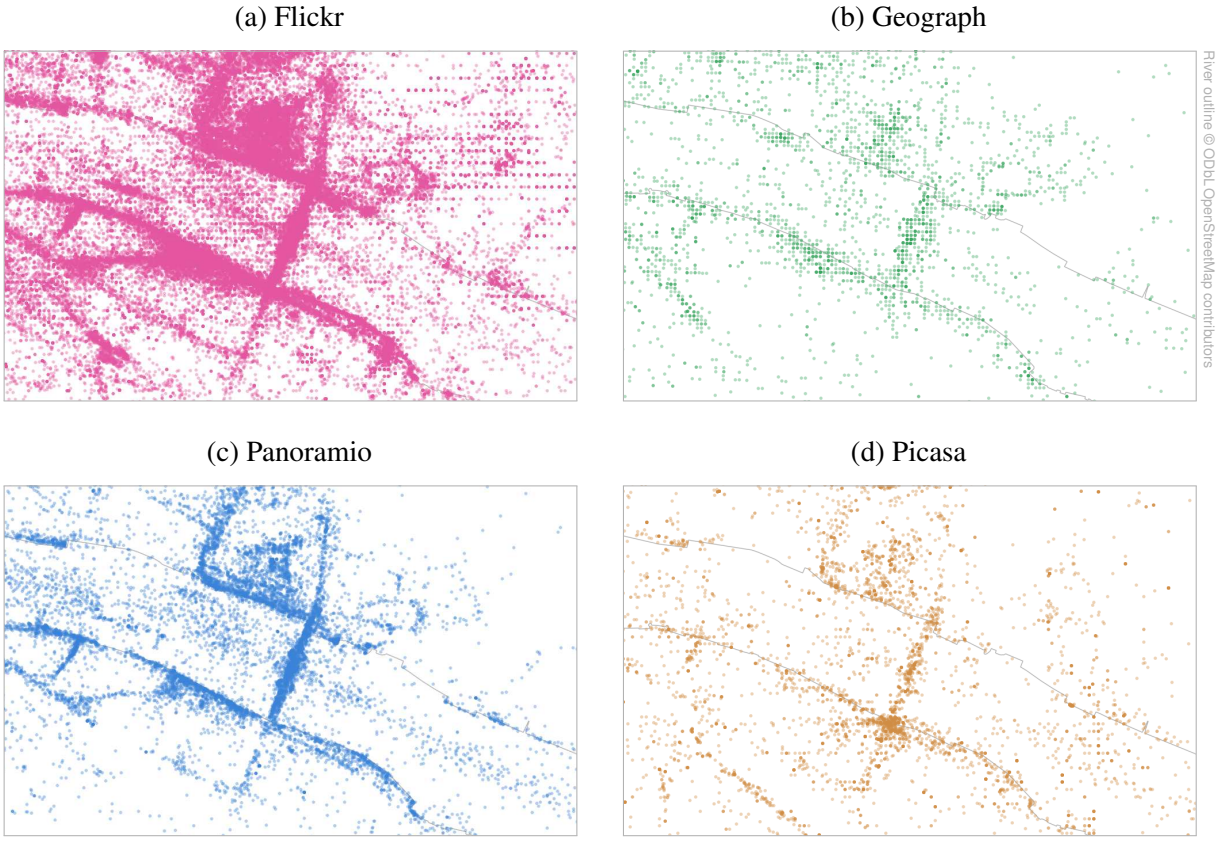


Figure 4.33: Grid effect in spatial distributions of photographs.

In three other sources of data the coordinates are either originally notated with two floating-point decimals or as a pair of degree-minute-second integer triads. Data granularity in the first case is equal to 0.000001° (rarely 0.00001° or 0.0001°), which results a grid of 7 by 11 cm and its multiples by 10 and 100. A degree second is equal to 0.000278°, thus forming a a grid of 19.24 by 30.89 m in the second case. A grid with these dimensions is the largest out of those that could be observed. Its size grows up to 30.89 by 30.89 m at the equator.

Figure 4.34 on the following page shows how often spatial coordinates of photographs belong to a degree-minute-second grid or a 10-meter OSGB grid, including their special cases (degree-minute grid and 100 m OSGB grid). As it can be seen from the maps and attached statistics, the proportions of coordinate types are diverse. Most of Geograph images belong to a 10-meter OSGB grid, thus defining dominant maximum spatial granularity for this dataset as ten meters. The vast majority of the photographs from three other sources do not belong to any grid – only 1.8% of Flickr records and 0.4% of those from Panoramio are mapped with granularity of 19 by 31 meters with use of degree-minute-second notation, while numbers of photographs on a 10 meter OSGB grid are negligible. As it can be observed from the distributions of black dots in Figure 4.34, coordinates rarely belong to special cases of both grids, which otherwise could mean that a proportion of photographs has been mapped with granularity of 100 meters or one degree minute (1154 and 1853 meters for longitude and latitude, respectively).

Hotspots in spatial distributions of geotagged photographs, which are arranged into grids, can be a source of bias if cells are bigger than the width of the attractiveness score window (the concept of a window is explained Figure 3.5 on page 50). Thus, if a street segment is located near the grid edge, it receives redundant ‘votes’ that belong to a large rectangular region – such ‘votes’ would be assigned to other edges in the same neighbourhood if original geotags had higher granularity. As the sizes of all discovered grids were smaller than previously extracted median displacement of images, no filtering of photographs by their relationship to detected grids was found necessary. The situation could be different if, for instance, Geograph photographs with low location accuracy were included into the cache (pre-filtering was possible by attribute `natgrlen` in the original database dump, which corresponds to ‘National Grid length’, or the size of the grid). Grids with larger sizes could be potentially found in Flickr too if `accuracy` was not defined as a *limiting agent* in API requests (see details on page 114).

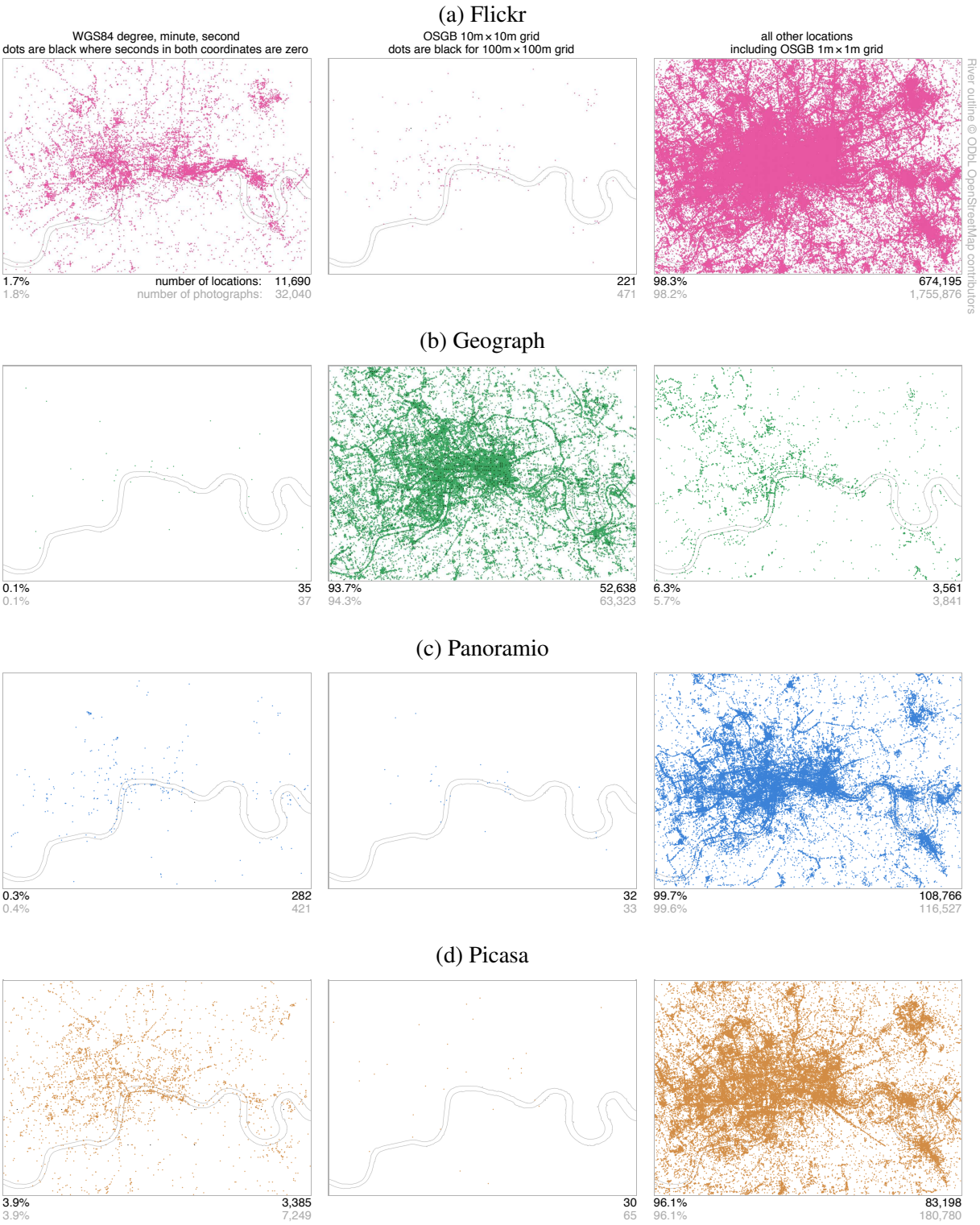


Figure 4.34: Unique locations of photographs by their proximity to coordinate grids. A location can concurrently be present in the first and the second column.

Hotspots that could not be explained by peculiarities in the granularity of original georeferences were found to form two groups. A part of these abnormal locations contained large numbers of photographs from a single user, while others had contributions from many authors, not necessary submitting a lot of images.

The origin of hotspots of the first type was suggested earlier in Subsection 4.3.1 – some users tend to mass geotag their works, placing tens of even hundreds of images on a map at once. In some of these cases the geotags can be accurate (e.g. if a person was attending a concert and took a hundred of shots from one spot), however, as data exploration has shown, these situations are rare. Mass-geotagged bundles of images usually contain scenes from different places, making a corresponding hotspot a potentially irrelevant 'vote' for street attractiveness.

Hotspots of the second type (those that are formed by many users) are probably accounted for the method of geotagging. Most of the photo-editing applications as well as web interfaces of some photo-sharing services provide search functionality and allow users define spatial coordinates for their images by typing a name of a locality. This can be a street, a building, an attraction, a district or even a city. When several users geotag their works using the same search term, the photographs end up having identical coordinates, referring to a centroid of an sought object. Such locations can be considered as less accurate than those assigned automatically by a camera or manually by a user when he or she drags an image onto a map. The bigger the sought object, the more redundant 'votes' appear near its centroid and the more bias in surrounding attractiveness scores is introduces.

Removal of both types of hotspots was found a reasonable step in data cleaning, which could make distributions of photographs better comply with requirement 3. It was nesessary to choose two threshold values, one for the maximum allowed number of photographs from a single user at a unique location and one for the maximum allowed unique users at a unique location. The smaller the values, the more inaccurately tagged photographs can be removed, but the more influential may be side effects of such filtering due to false positives. Given the complexity of the data and impossibility to apply traditional statistical analysis for decision-making in such situation, judgement was made by means of visual analytics. Unique locations of images broken by maximum number of photographs from a single user and number of unique users are shown in Figures 4.35 and 4.36.

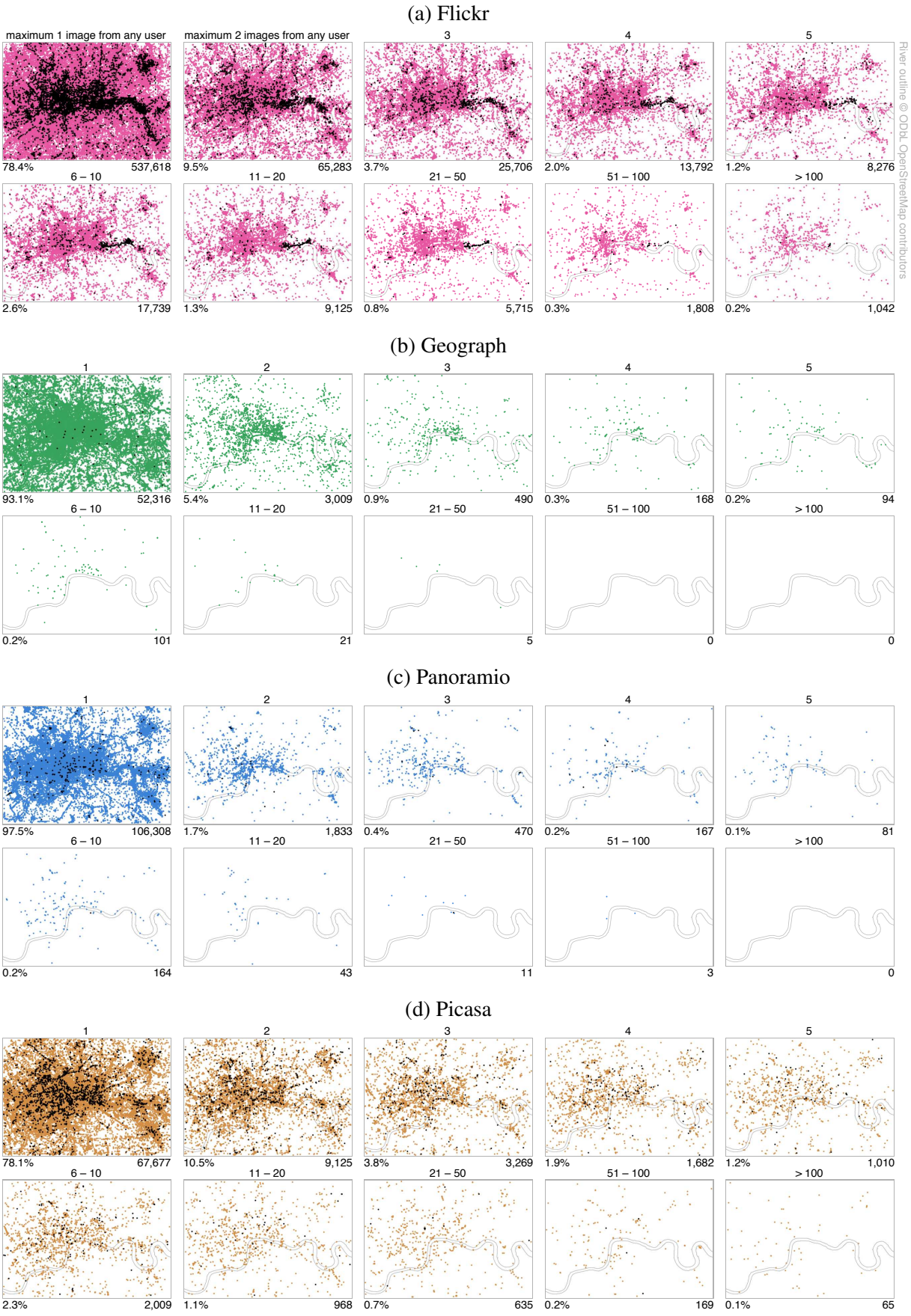


Figure 4.35: Unique locations of photographs by maximum number of records from a single user. Black dots correspond to coordinate that belong to a degree-minute-second grid.

Figure 4.35 on the facing page helps see how often users of different photo-sharing services apply identical geotags to multiple images. The cases of mass geotagging occur more often in Flickr and Picasa, being significantly less common in Geograph and Panoramio. Over a thousand of unique locations in Flickr contain over a hundred of photographs from one user, which corresponds to 0.2% of all unique locations total (in other words, that is every 500th unique location). Arrangement of locations with high numbers of photographs appear to repeat general distributions of records in all four cases – the more popular the region, the more cases of mass geotagging take place.

Because spatial coordinates have different original granularity, one can expect to see a shift towards higher numbers of photographs at some locations in cases when geotags belong to one of the discovered grids. This trend was not found significant during the analysis and was considered as negligible. But, for example, if a proportion of photographs at the degree-minute-second grid was high in any of the datasets, it would be necessary to define a separate threshold for images that belong to this grid. Contribution from Flickr user *MadPole* can help depict this potential situation (see Figure 4.20a on page 139, first row, second column). Over 10,000 of their geotags were automatically assigned by the smartphone in a degree-minute-second notation, placing images onto a 19 by 31 meter grid. As a consequence, one can observe a significant number of black dots in the second row of small multiples in Figure 4.35a around the area of this user's activity. This means that that user *MadPole* alone creates a shift towards more photographs per unique location. If such cases were more common, it would be necessary to distinguish between coordinates with different granularity and apply different filtering thresholds to avoid mass exclusion of valid votes from some very active users.

Figure 4.36 on the following page shows how frequently different photographers share identical locations. Because coordinate plane is not infinite for any initial spatial granularity, it is normal to expect overlaps, especially in popular areas. When coordinate displacement is even (i.e. there is no bias in geotags), numbers of overlaps and their volume are expected to fluently decrease or increase – individual standalone outliers are rather unlikely. This is not the case in all four considered datasets, as the visualization demonstrates. For example, Flickr distribution contains over a hundred unique locations with contributions from fifty or more users. Most of these locations are standalone, i.e. do not have any other popular neighbours.

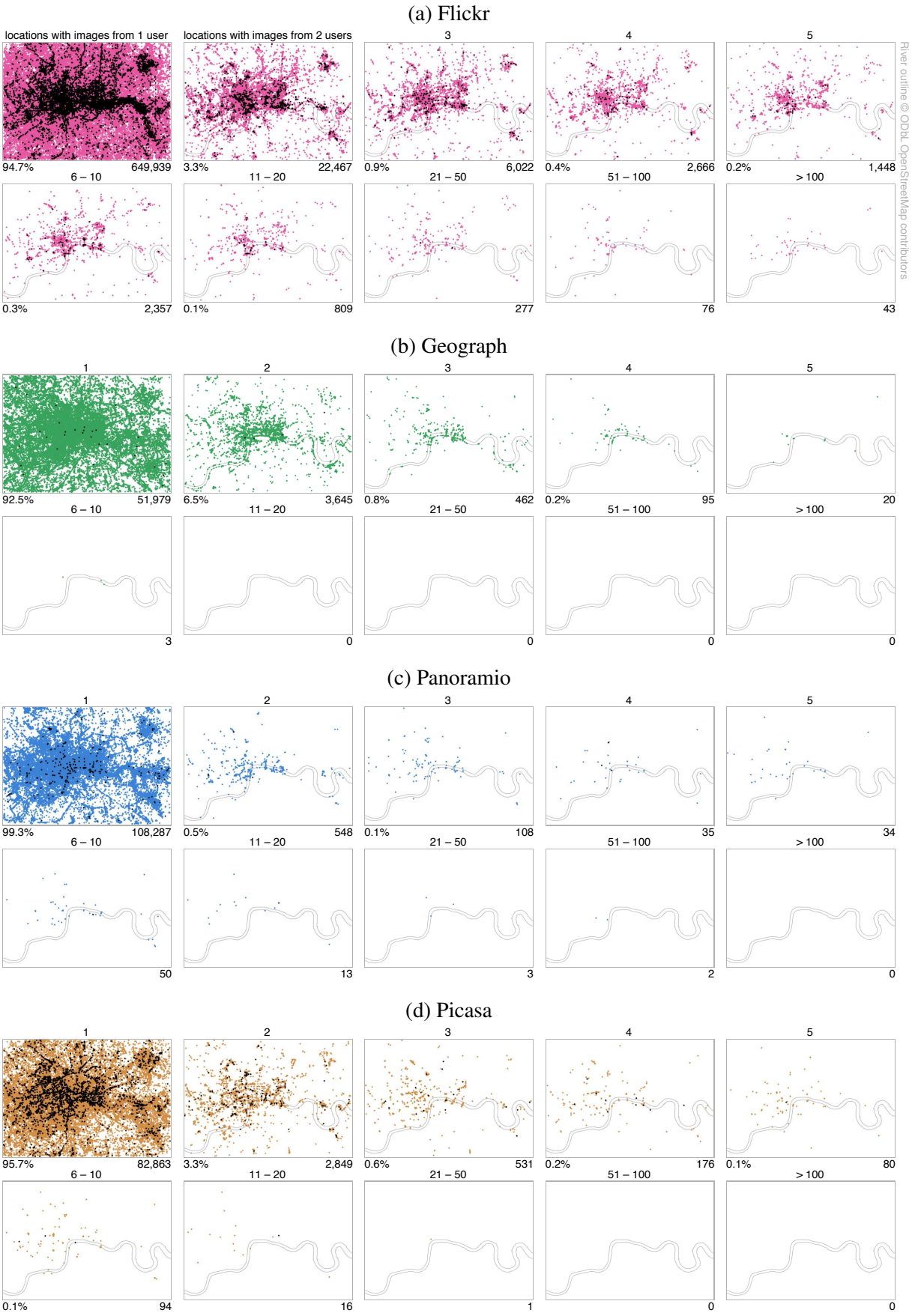


Figure 4.36: Unique locations of photographs by numbers of users. Black dots correspond to coordinate that belong to a degree-minute-second grid.

Detailed exploration of a sample of these coordinates confirms that they are located in the centroids of popular places (e.g. Trafalgar square, Buckingham Palace, Tate Modern, Tower Bridge, St. Paul's Cathedral). This proves a hypothesis that such hotspots are introduced when photographers use place search to attach their works to a map.

If coordinates of photographs have different granularity, it is likely that ones that belong to a sizable grid may also form hotspots of the second type. If a filtering threshold is low, it is probable that these locations can be confused with search-related hotspots, and such situation may bring a need to treat various locations differently. This effect was not discovered in the data, which is partially demonstrated in Figure 4.36 (see the distributions of the black dots, corresponding to locations on a degree-minute-second grid).

The probability of location overlaps depends not only on the granularity of the spatial coordinates and a relative popularity of an area, but also on the overall number of users – the more popular a photo-sharing website, the more locations may contain contributions form two or more individuals. This suggests that a filtering threshold should be different at least for Flickr, which contains over ten times more photographs than any other dataset in the considered region.

In this research it was decided to use the following thresholds for hotspot filtering:

> *10 photographs from a single individual for hotspots of the first type*. If there are more images at a location from the same user, all are considered as rejected due to mass geotagging, which is likely to result a significantly displaced 'vote'. In cases when the number of photographs is less or equal to a threshold, only one image is kept. Such additional data reduction is valid because of an earlier decision to count attractiveness scores by looking at numbers unique active users instead of the numbers of photographs in the neighbourhood. Filtering of hotspots of the first type does not affect contributions from other users with photographs at a given location. For example, if a location contains a hundred photographs from user A and one photograph from user B, the latter one persists in the data.
>
> *5 unique users in Flickr and 2 unique users in other datasets for hotspots of the second type*. More popular locations are considered as those that contain photographs, which are geotagged using place search. All images in such hotspots are marked as rejected.

Thresholds for both types of hotspots are subjects to a further discussions, and detailed analysis of a large sample of hotspots may reveal that the above numbers should be changed to increase the reliability of filtering. It is believed, however, that such fine tuning may not significantly affect the quality of attractiveness scores. As it can be observed from the statistics attached to Figures 4.35 and 4.36, the proportions of locations in bins on both sides of the thresholds are very small, so picking neighbouring values may not cause substantial distortions in data. Importantly, any reasonable thresholds will exclude evident outliers, which contain the largest amount of bias.

Visual analytics approach was found useful not only for filtering justification, but also for confirming the removal of hotpots. This is demonstrated in an example in Figure 4.37. Changing *alpha* (transparency) of circles representing locations of photographs and adding *variance* to coordinates (a random component with a given standard deviation) helps see hotspots in the original distributions of images and their absence in the filtered versions of the datasets, here in Flickr.

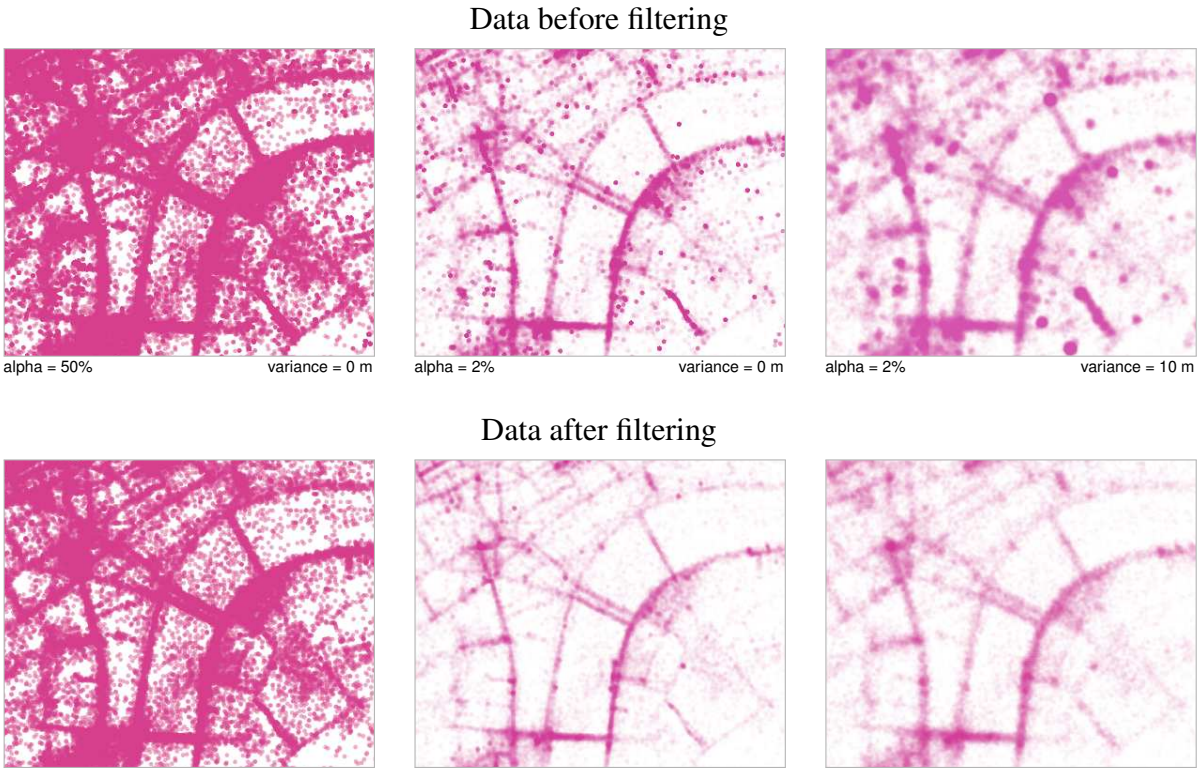


Figure 4.37: Confirmation of hotspot removal in Flickr dataset.

4.3.4 Event detection

Analysis of temporal coordinates of photographs in Subsection 4.3.2 revealed that user activity is influenced by local and global events. The 'votes' submitted during concerts, carnivals, protests or other special occasions are likely to be a soure of bias in street attractiveness scores as they do not help describe the attractiveness of the environment on an average day. This contradicts with requirement 10 for a *model photographic collection* (page 38).

Detection and filtering of local events is not practical on a 'global' scale, i.e. by considering temporal peaks in activity in the whole cached region and excluding days when unexpectedly many users take photographs. This is difficult because of the general unevenness of temporal distribution (Figure 4.27 on page 150) and little influence of local anomalies on general trends. The bigger the region, the more problematic this approach becomes.

The question of event detection in crowd-sourced photographic datasets has been previously studied in a number of research projects (e.g. Quack, Leibe and Van Gool 2008; Gao, Hua and Jain 2011; Andrienko et al. 2012). All works share a similar approach: space is tessellated into small regions, which are then independently tested against the existence of peaks in temporal distributions. Inspired by the successful use of Voronoi spatial clustering for event detection (Andrienko et al. 2010), it was decided to adopt this particular approach for exclusion of 'votes' that violate requirement 10. Unlike static clusters with predefined shape and size, Voronoi polygons are constructed with respect to local spatial arrangements of the data, increasing the likelihood of logical grouping of the records. The method is robust and scalable by design.

Experiments with Voronoi tessellation in this research were performed with use of a third-party GIS toolkit, kindly provided by Professor Gennady Andrienko from Fraunhofer Institute, Germany. A screenshot of the process is shown in Figure 4.38 on the next page. Several versions of pre-filtered Flickr, Geograph and Panoramio distributions were exported into CSV files, which were then loaded into 'Common GIS research' software. Clustering was done with one of the built-in modules for geographic computations. After the Voronoi polygons were generated and saved as XML files, they could be easily attached to the datasets in the main database for further analysis by means of the Dataset Abstraction Framework.

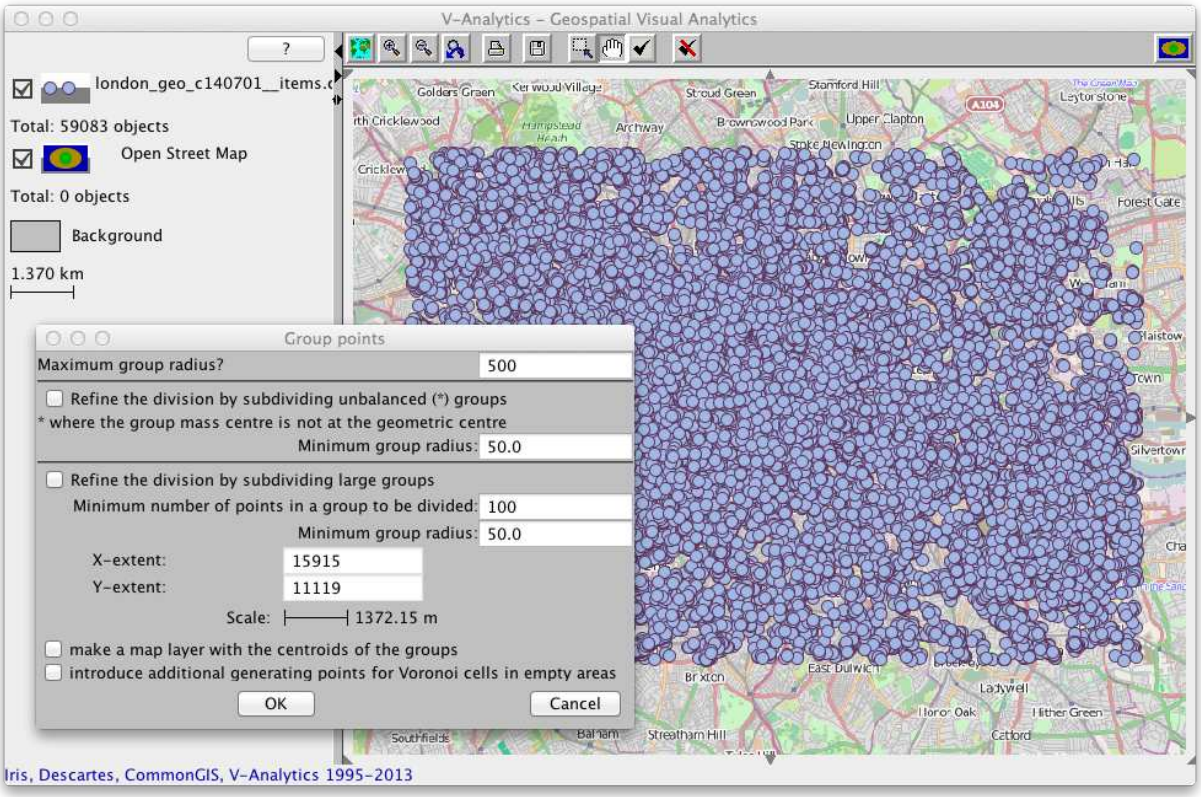
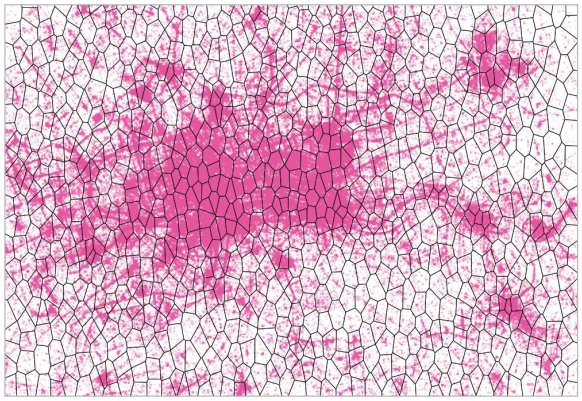


Figure 4.38: Spatial clustering using Voronoi algorithm and 'Common GIS research' software.

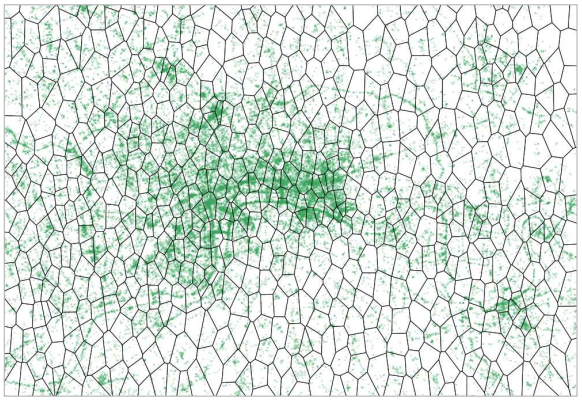
During the consultations with Gennady Andrienko and exploration of the results of clustering with various configurations, it was concluded that maximum and minimum group radius of 500 and 50 meters were optimal for the chosen purpose. When clusters are made too small, many popular tourist attractions, squares and buildings become split into several parts, which makes event detection more difficult. On the other hand, clusters of larger size often include multiple venues, where events take place independently. This also potentially complicates the process of filtering and reduces the quality of the outcome.

The result of Voronoi clustering for the latest versions of three out of four photographic datasets is shown in Figure 4.39 on the facing page. The cached Picasa distribution was not tessellated as it was rejected at the stage of data gathering. Despite a significant difference in overall popularity of Flickr, Geograph and Panoramio, the numbers of obtained Voronoi polygons were similar, in correspondence with the configuration (910, 747 and 693, respectively). A summary of daily user activity within the most popular clusters is also shown on the next page, in Figure 4.40.

(a) Flickr



(b) Geograph



(c) Panoramio

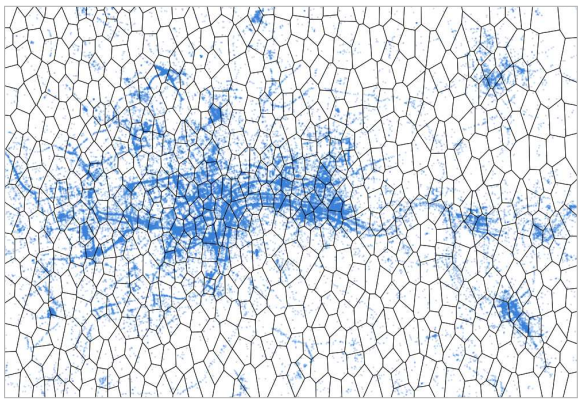
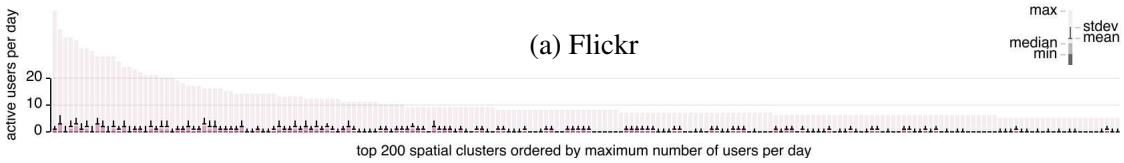


Figure 4.39: Spatial clusters with maximum radius of 500 meters, obtained using photographs that passed previously discussed filters.



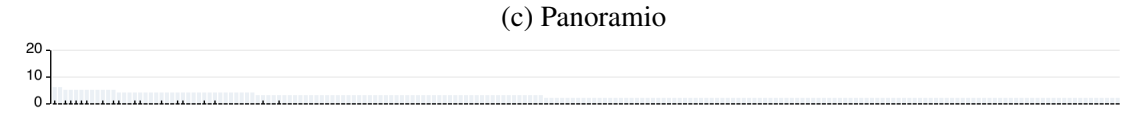


Figure 4.40: Statistical measures of daily user activity in 200 most popular spatial clusters.

As it can be seen from the bar charts, mean, median and standard deviation of numbers of daily active users are generally extremely low. Almost all of these measures in Geograph and Panoramio are approaching zero. Maximum values of numbers of daily active users in all three datasets appear to be much more distant from the mean than the difference between the mean and the minimum. This describes the distributions as positively skewed or with outliers.

Figure 4.41 on the next page reveals the details of user activity in four manually picked clusters to demonstrate local temporal profiles. Trafalgar Square and Bank of England are among the most popular locations in all three datasets. A section of Regent's canal near Millennium park was selected as an example of a cluster with an average activity, and Olympic Stadium was chosen for being significantly influenced by the Games in 2012. Flickr distributions demonstrate higher levels of daily activity within the clusters, as it can be expected from the greater popularity of this photo-sharing service compared to Geograph and Panoramio. Besides, this dataset can be characterised with more occurrences of the outliers. The biggest number of active users per cluster per day in the whole collection of data is observed in Flickr on the first of April 2009 near Bank of England. This extreme peak is influenced by the massive G-20 London summit protest, which happened on the same day (BBC 2009*b*). Manual exploration of the photographs confirmed this connection, and a study of other local peaks in a number of clusters also demonstrated their relationship to various events.

Removal of peaks within clusters was considered as a positive influence on the photographic distributions – this could make attractiveness scores less biased by the local events, especially in Flickr data. It was found logical to apply a slightly different approach for filtering compared to one that was used to exclude global events from the datasets (see Subsection 4.3.2 on page 147). If days with detected peaks were completely removed from the data, this could cause a negative 'over-filtering effect', as the numbers of events were likely to be different in various clusters. For example, if fifty days were completely excluded at Trafalgar Square and only five days at Bank of England, this would be identical to looking at these two clusters for unequal time periods. Such situation could potentially distort the proportions of scores in regions like these, making roads in clusters with more events less attractive than they 'deserve'.

Instead of complete exclusion of all contributions during days that are identified as events, it was proposed to suppress user activity by removing entries from only a proportion of photog-

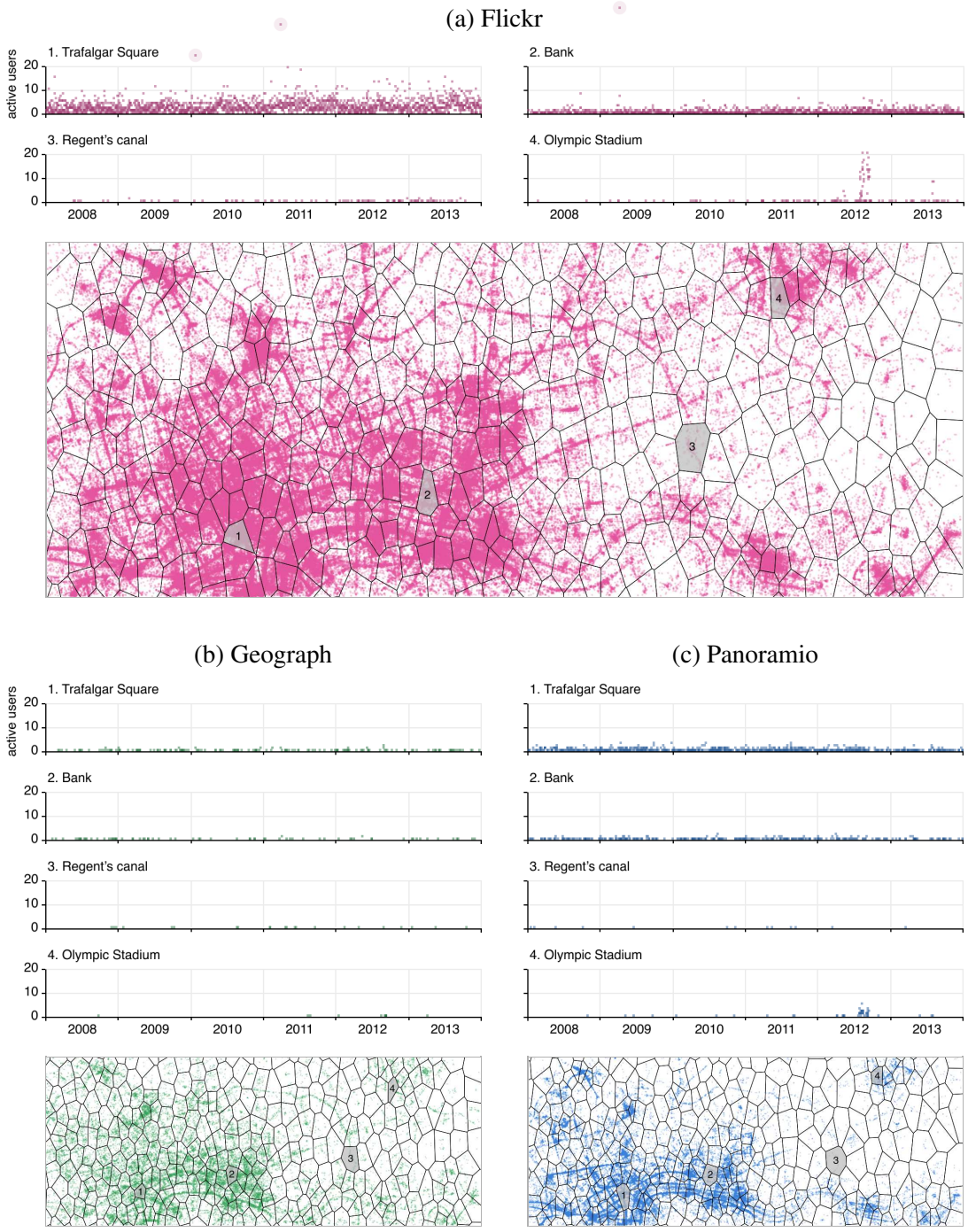


Figure 4.41: Examples of user activity over time inside spatial clusters.

raphers. This approach suppresses the peaks, but does not make popular clusters potentially underrepresented. When choosing users that should be removed, preference can be given to those that contribute to a cluster only during an event, thus increasing the likelihood of reduction of numbers of 'votes' in attractiveness scores. Although some photographs that are left after filtering may be still depicting events, their proportion in a dataset reduces significantly.

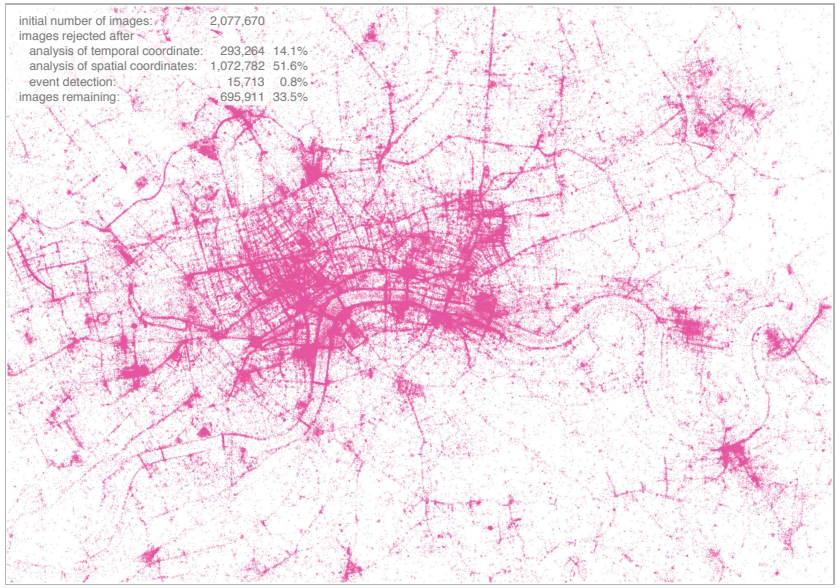
A strategy for local event filtering in this project was chosen in a series of experiments, similar to those demonstrated in Figure 4.28 on page 152. Because of differences in levels of overall cluster popularity, it was found reasonable to vary a threshold based on the local statistical measures rather than to use a fixed value (e.g. 10 users per cluster per day for Flickr, 2 – for Geograph, 5 – for Panoramio). Seasonal variations and long-term shifts in popularity of photo-sharing services were almost invisible in local distributions, which made it possible not to change the thresholds over time within the clusters. The final formula for a threshold was defined as a sum of a median and a standard deviation of local daily user count plus a fixed gap. The gap of three users per cluster per day was introduced to avoid the exclusion of all contributions in the least popular clusters and also to allow for some natural fluctuations that do not necessary imply the existence of an event. With this threshold it was possible to detect and process 1,846, 8 and 26 cluster-events in the latest Flickr, Geograph and Panoramio datasets, respectively.

4.3.5 Results

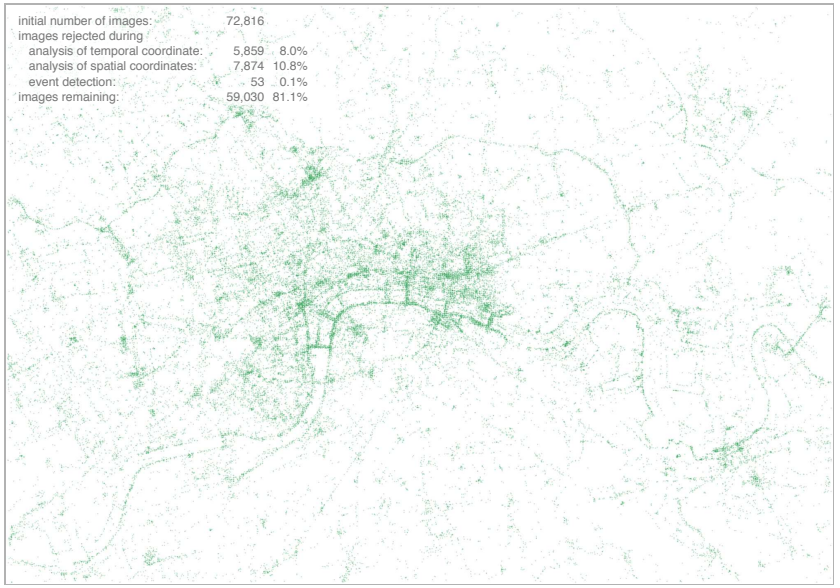
Figure 4.42 on the facing page shows how the filtering methods, introduced above, changed the distributions of records in the latest versions of Flickr, Geograph and Panoramio datasets. Looking at Figures 4.18 on page 136 and 4.27 on page 150 can be useful to see the difference.

Flickr, being a general-purpose photo-sharing website, was affected most of all, with only a third of images remaining. Geograph and Panoramio datasets had less cases of mass geotagging, biased spatial coordinates and increased user activity during local events. This meant that they were originally closer to a definition of a *model photographic collection*, given that all photographic collections went through the same filtering procedures.

(a) Flickr



(b) Geograph



(c) Panoramio

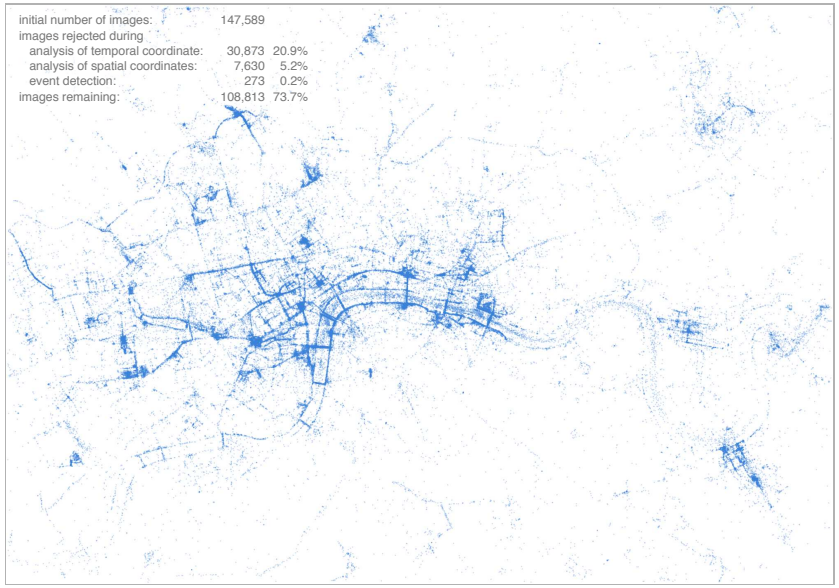


Figure 4.42: Images that remained in the latest versions of photographic datasets after the distribution-based filtering.

4.4 Photo assessment survey

In accordance with the workflow for the analysis of photographic datasets (page 42), it was necessary to assess the contents of images after exploring their spatial and temporal distributions. Given the list of photo-sharing websites chosen in Section 4.1 on page 107, this study was only relevant to Flickr, Geograph and Panoramio collections, because Picasa data was found incompatible with the requirements to a *model photographic collection* at the stage of their gathering. Photo content assessment was performed in a form of a public online survey, where volunteers classified random samples of images using seven criteria. The design of the questionnaire is justified and explained on pages 45–47. This section contains the description of the data gathered during the survey and also discusses the findings about assessed photographic collections.

4.4.1 The data

Call for participation in the survey was opened on the 7th of December, 2012. The link to the website (`http://www.photoassessment.org/`) was several times published in social networks and related forums, circulated via email and also shared by a number of respondents.

The survey contained 901 images – 300 sampled from Flickr, Geograph and Panoramio plus one manually picked introductory photograph. The latter was always located in beginning of each participant's random queue, but its classifications were ignored. This photograph, the content of which was deliberately easy to assess, both made people familiar with the interface of the survey and worked as a simple protection from those who were not interested in contributing sincere answers.

The samples were randomly chosen from the latest versions of cached datasets at the time of the launch of the survey, but preference was given to images with attached EXIF metadata to enable their further analysis. All images were reported to be taken between 2008 and 2012. Importantly, sampling was done independently from filtering, enabling general conclusions about the chosen photo-sharing services.

By the time the survey was closed in the end of March 2013, it was viewed by 608 volunteers, 359 of whom classified at least one image excluding the introductory photograph. 8,434 gathered responses enumerated 49,285 classifications (answers to individual questions). Every photograph was covered with responses from at least eight participants.

4.4.2 Exploration of responses with visualization

Given a subjective nature of the responses and a need to understand the collected data in detail, it was decided to explore the results of the survey with an interactive visualization. Its interface, shown in Figure 4.43, was designed around the novel concept of a 'survey response glyph', a detailed explanation of which can be found in Subsection 3.3.4 on page 101 and in Kachkaev, Wood and Dykes (2014). With linked views, details on demand, and interaction as in Figures 4.44 and 4.45, it was possible to navigate through all the collected data in an efficient way and to discover a number of important patterns that could be missed if the responses were aggregated in a traditional way and only then analysed.

Following the recommendations described in Subsection 3.3.4, two survey response grids were constructed to become a base for the glyphs. Both are shown in Figure 4.46 on the next page.

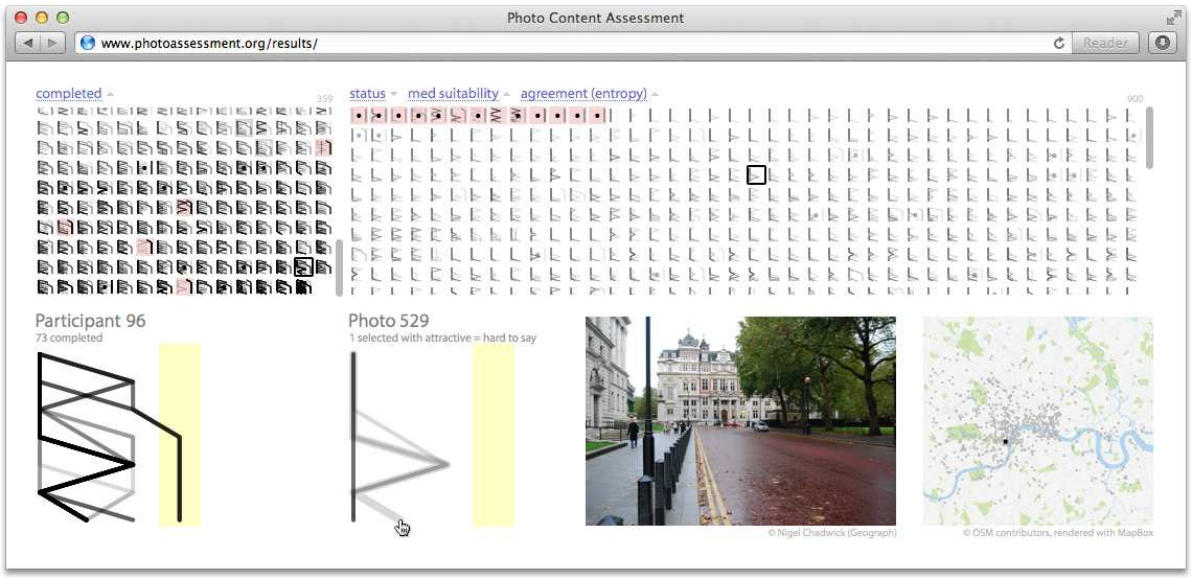


Figure 4.43: The interface of the survey analysis tool allowing navigation through collected responses with use of glyphs, linked views and interaction.

Figure 4.44: Multilevel sorting of the entity lists.

Figure 4.45: Cross-highlighting relationships between groups.

		1	2	3	4
real photo?	1	no	h. t. s.	yes	n. a.
outdoors?	2	no	h. t. s.	yes	n. a.
daytime?	3	night	h. t. s. / twilight	day	n. a.
temporal?	4	no	h. t. s.	yes	n. a.
people?	5	no	h. t. s.	yes	n. a.
by pedestrian?	6	no	h. t. s.	yes	n. a.
attractive?	7	no	h. t. s.	yes	n. a.

(a) Standard ordering of categories reflecting the original order of questions and response options in the survey.

		1	2	3	4
real photo?	1	**yes**	h. t. s.	**no**	n. a.
people?	2	no	h. t. s.	yes	n. a.
outdoors?	3	**yes**	h. t. s.	**no**	n. a.
daytime?	4	**day**	h. t. s. / twilight	**night**	n. a.
temporal?	5	no	h. t. s.	yes	n. a.
by pedestrian?	6	**yes**	h. t. s.	**no**	n. a.
attractive?	7	**yes**	h. t. s.	**no**	n. a.

(b) Purpose-oriented ordering of categories (alterations in bold). Answers ordered so that photographs showing attractive walkable areas are represented with a straight vertical line on the left.

Figure 4.46: Survey response grids.

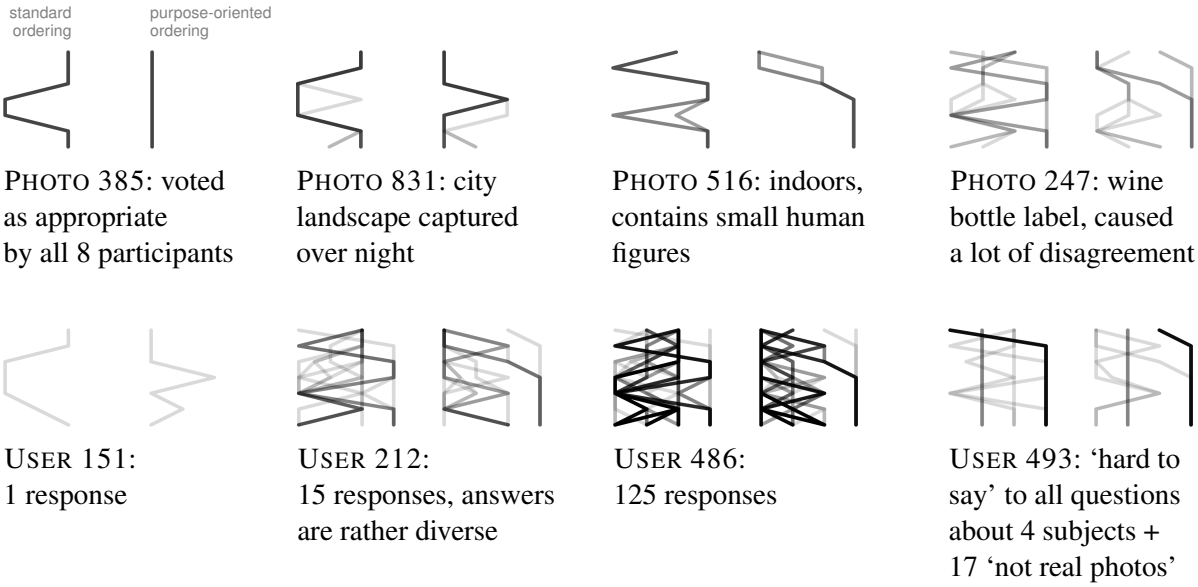


Figure 4.47: Glyphs representing responses grouped by photographs (samples) and users (participants). Opacity of a single line is 15%. Vivid patterns in groups denote a high level of agreement among responses.

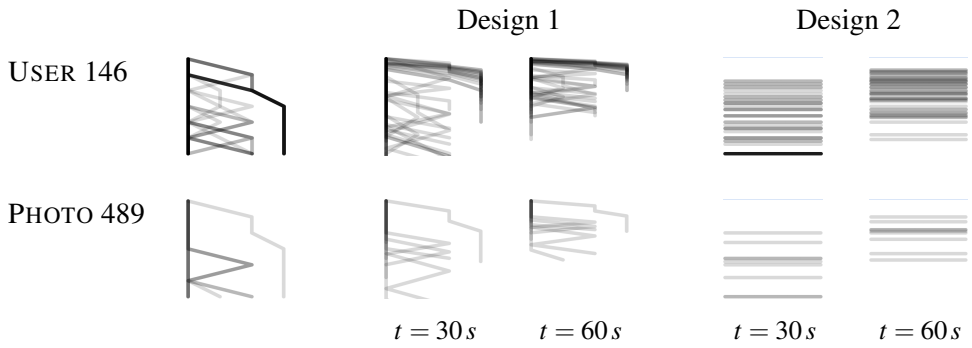


Figure 4.48: Time scaling of the response glyphs.

Despite that the maximum number of categories per question was five (see *daytime*), the grid was limited to four columns by merging ‘twilight’ and ‘hard to say’ responses due to their semantic similarity in the context of the analysis. Although the answers to all questions were not numeric scores, they could be still considered as ordinal; ‘n. a’, being an exception (nominal value), could be optionally highlighted to support glyph interpretation.

Having one response grid should be usually enough for most surveys, however, in the given case it was reasonable to introduce *purpose-oriented ordering* (Figure 4.46b) of the grid in addition to *standard ordering* (Figure 4.46a). With this design the photographs that depicted attractive walkable areas were represented with straight vertical lines on the left, which could be instantly decoded. The order of questions in the *purpose-oriented grid mode* was also changed by moving one about people to be the second. This decision was determined by conditional question set within the survey: the question about the spatial environment of a photograph when the answer was ‘indoors’ disabled remaining questions.

Figure 4.47 on the facing page contains examples of some response glyphs.

Where lines showed repeated similar responses made by participants (standard ordering), they became a useful indicator of possible response insincerity. If a respondent clicked on all available controls at the same position, for example in an attempt to skip through the questionnaire rapidly, their actions became represented as vertical lines, which became darker for persistent behaviour. The fact that in the survey the answer ‘no’ to the first question disabled the remaining ones allowed participants to proceed to subsequent photographs even faster, in only two clicks. Such behaviour was also easily detectable by examining the glyph patterns. Because six of seven questions become ‘n.a.’ when an image was classified as not a photograph, the response became depicted as a ⅂ shape. A darker line with this shape was a clear indicator of unwanted behaviour that required removal. User 493 (Figure 4.47) is an example of insincerity: after giving a few possibly considered responses, he or she answered ‘hard to say’ to all questions as a means of proceeding to the next photograph quickly and then discovered the first ‘no’ shortcut.

The main advantage of the purpose-oriented ordering over the standard one was to support selection of the photographs most suited to characterising street attractiveness. With a universal

and straightforward rule, which in this particular case was 'the further from the left the less suitable', this glyph type allowed immediate detection of whether a photograph was good for the chosen research purpose and estimating the degree of its unsuitability. This statement is exposed in Figure 4.47 on page 178 (photos 385 and 516).

As the amount of time it took a participant to complete all answers about each subject was recorded, it was possible to merge this response attribute into a glyph. Such combination of data revealed participants that were unusually quick, and also enabled a comparison of the overall difficulty respondents had in classification of different photographs. Two glyph designs incorporating the duration of the responses were developed and used, as shown in Figure 4.48 on page 178.

The first design was based on the idea of scaling lines vertically according to the response time of users to each question. The glyph in this case has similarities to the *DriftWeed* visual metaphor proposed by Rose and Wong (2000). With a given time window t as a parameter the responses completed in t seconds had a height equal to a 100% of the glyph size, others were shorter or longer. As there were no data about how much time it took a participant to move each control individually, the shapes were stretched evenly. This scaling revealed some expected patterns like the correlation between the number of available questions and the response duration and also confirmed insincerity in the behaviour of some participants.

The second design showed the durations of the responses with horizontal lines, thus displaying only one attribute per response in a glyph. In this case a baseline on top of the glyph was added to serve as a zero reference point and was coloured differently to the responses. Despite the simplicity of the view it allowed visual estimation of the complexity of the classification of the subjects and also the differences in the performance among the participants.

As the glyphs were drawn in a software environment, they could be made responsive to mouse hovers and clicks. Mouse movements over the lines highlighted the context and showed additional information on demand (see Figure 4.43 on page 177). Clicking on a line in a glyph where the responses were grouped by subject opened the details of a corresponding person in the paired view; the opposite applied to the responses grouped by participants. To make the interchanges between glyphs smooth and clear, animated transitions between their states were

added (Heer and Robertson 2007). This allowed tracking of individual responses when toggling between various glyph designs as well as whether different groups of responses shared the same subjects or participants.

With use of glyphs, all collected survey data could be compactly displayed on a single screen with very little loss of detail, as it is shown in Figure 4.49. Multilevel sorting and real-time interaction together with some introduced auxiliary symbols and representations for the glyphs essentially facilitated the analysis of the volunteers' input. Examples of customised lists of responses grouped by survey participants and photographs are shown in Figures 4.50 and 4.51 on the following page.

The findings derived from the visual exploration of the data can be divided into two main groups: the observations that helped maintain the survey process and the task-related conclusions about the photographic data.

The first benefit of using visualization was in the improvements to the queue forming algorithm, which it suggested to make. After running the survey for one month and assigning the images to the new participants randomly, it became visible that this approach did not deliver a smooth coverage of the photographs with the responses. Unevenness in the colour of the

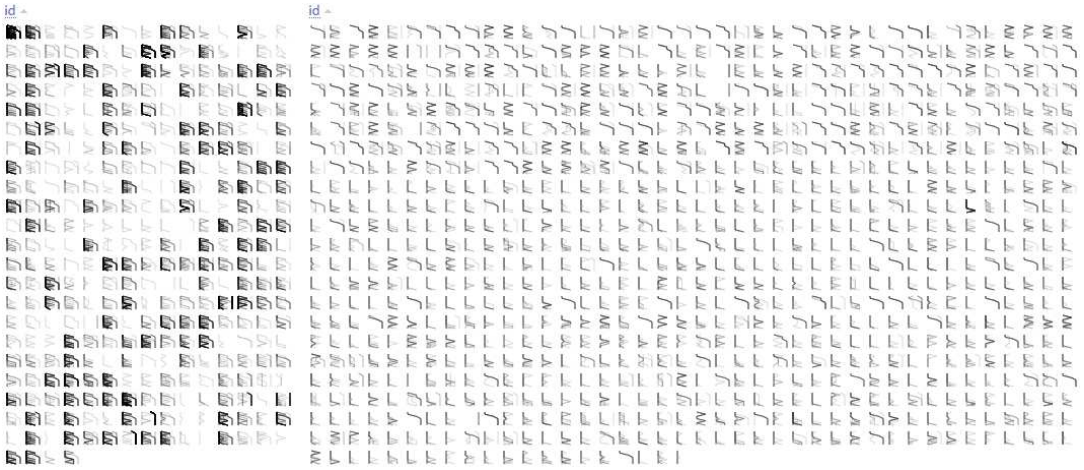
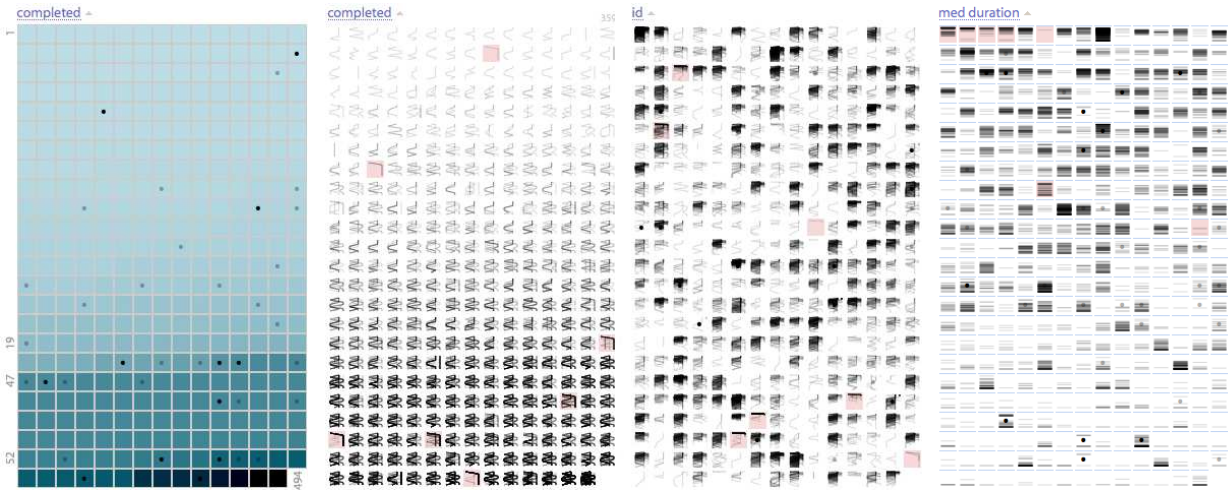


Figure 4.49: Lists with all survey response glyphs (purpose-oriented ordering) with *left:* grouping by users (participants) and *right:* grouping by photographs (the subject of the survey). Both lists are sorted by the entity ids, which orders users by their joining date and splits the list of photos into 3 equally sized groups according to their source (Flickr, Geograph and Panoramio), but keeps the order within those groups random.



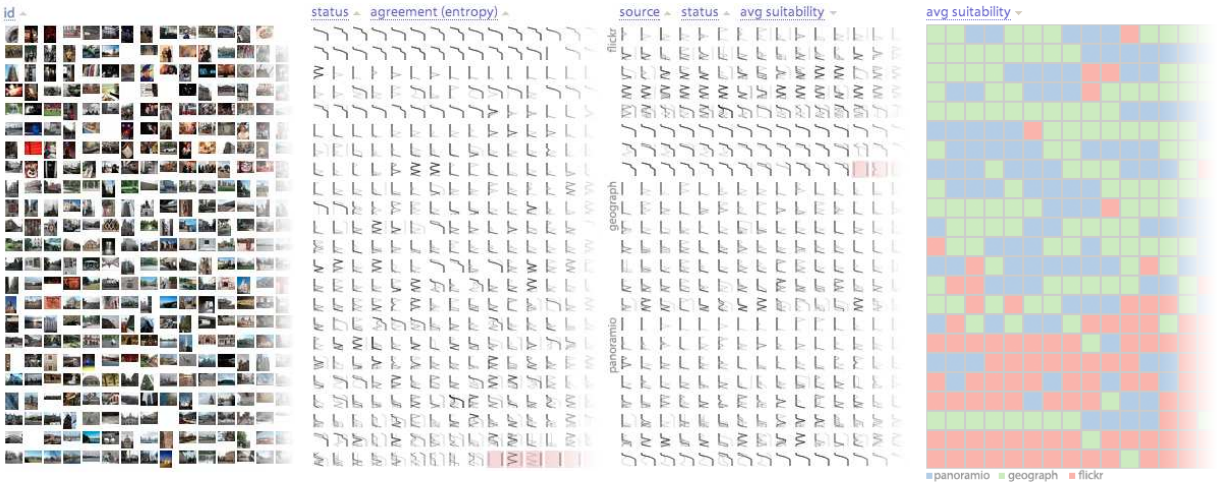
(a) Elements coloured by the number of completed responses; • – photoservice API faults.

(b) Standard grid glyphs; red cells represent potentially insincere participants.

(c) Glyphs drawn in time mode 1 (30 *s*, standard grid) and sorted by their id.

(d) Glyphs drawn in time mode 2 (30 *s*) and sorted by the participants' median response duration.

Figure 4.50: The list of survey participants with different representations and orderings.



(a) Photo previews.

(b) Purpose-oriented grid glyphs sorted by the amount of agreement between responses.

(c) Same glyphs ordered by the source of photographs, their status and the average suitability.

(d) Colouring of items by source and sorting by the mean of suitability.

Figure 4.51: The list of survey subjects (photographs) with different representations and orderings. Not all elements are shown for compactness.

glyphs in the list on the right-hand side led to a change in the photo assignment algorithm – preference was given to those images that were lacking data. The photographs with the same priority were still randomly distributed among the participants, which guaranteed no bias to evolve with this alteration. Afterwards, by looking at the data visualization it was discovered that counting only the successful responses for queue prioritisation was not a universal solution. The photographs that were recently deleted from a photo-sharing service but not yet manually excluded from the survey were still shown to all new participants. They could not collect any new valid responses, but retained the highest priority. Seeing errors instead of images at the beginning of the survey could discourage the participants and reduce the numbers of submitted responses. Visual analytics helped reveal and solve this problem too – the numbers of photo API faults were added to the priority counting algorithm in cases when problems occurred more than once. While the first decision regarding the queue forming could be made after seeing the statistics in a spreadsheet or by drawing a simple bar chart, further improvement would be hard without an interactive visualization of the entire dataset.

Glyphs helped track the level of user engagement with the survey. Ordering the participants by the numbers of assessed photographs revealed a clearly visible plateau of similar values around 50 responses per user (Figures 4.50a and 4.50b) despite it was expected to see an inverse log distribution. A very similar level of engagement of nearly a quarter of participants was explained with the presence of a photo queue in the survey interface (see Figure 3.4 on page 48). Although the description of the survey on the welcome page clearly stated that the assessment of any number of photographs was not compulsory, progressing towards the imaginary end of the queue was considered by many participants as a goal and provoked curiosity. Thus, the visualization on one hand engaged users to submit more responses and on the other hand helped discover and evaluate the effect from this engagement.

Importantly, survey glyphs were found a powerful instrument for analysing the participants' behaviour. Placed next to each other, they enabled the detection of anomalous response patterns such as massively submitted insincere answers. With the help of the survey data exploration tool, input from seven users was rejected based on the shapes of the corresponding glyphs (Figures 4.50b, 4.50c and 4.50d) – most of these participants were answering 'no' on the first question to quickly move along a queue of photographs with minimum effort.

The main reason for conducting the photo content assessment survey was a desire to know the proportions of potentially valid 'votes' for street attractiveness in the chosen image sources. With use of glyphs it was possible to see the level of agreement in the participants' opinion and to compare the content of photographs in the samples. Ordering the list of purpose-oriented photo glyphs by entropy (Figure 4.51b) revealed that the agreement in the answers given by different participants was generally high – the majority of glyphs contained a single bold line, which represented the most common answer among minimum eight of them. Rare cases of images that the participants could not classify in a similar way were mostly close-ups, photographs of artworks or weather events (e.g. a rainbow during the sunset). Less peculiar scenes such as photographs taken on a street, during an event or inside a building usually did not cause major disagreement, which was a sign of a good reliability of the collected data.

The differences of the chosen samples of images could be easily examined with use of purpose-oriented photo glyphs. Ordered by the source and the average suitability (Figure 4.51c), these glyphs revealed the proportions between the 'ideal votes' (|) and the least appropriate ones ( ), also showing all existing intermediate cases. It was found that among the images from Flickr there were significantly more indoor photography, pictures taken during events or containing people. Such cases were less common in Panoramio and Geograph samples, which looked relatively similar. Panoramio sample appeared to have a slightly higher proportion of indoor photography compared to Geograph, but the difference was insignificant.

Dissimilarity in the content from Flickr relative to two other photographic sources could be also recognised with an alternative representation of the list, shown in Figure 4.51d on page 182. When survey subjects were coloured by their origin and sorted by the mean of their suitability (i.e. by the 'distance' of a glyph from |), most of the rectangles representing Flickr appeared to be in the bottom of the list. This showed that the images from this source were in general violating more requirements to a *model photographic collection* (page 38) compared to those from Geograph and Panoramio.

All collected survey data can be explored at `http://www.photoassessment.org/results/`. A video with an explanation of the response glyph concept and a demo of the discussed interactive visualization is available at `http://vimeo.com/90299533`.

4.4.3 Aggregation of survey results

Glyph-based exploration of subjective photo content classifications clearly showed that not all assessed images could be used as 'votes' for street attractiveness. Given a reasonably high amount of agreement in the contributed human opinions, it was possible to aggregate them, thus supplementing every image with seven fixed attributes that matched the most common answers in the questionnaire. A distribution of categories extracted from mode user responses is shown in Figure 4.52.

Aggregated summary of image classifications confirmed the findings discussed earlier and allowed important observations to be made. Given that the samples were random and sufficiently large (300 images per photographic source), these observations could lead to a number of reliable general conclusions about the applicability of the chosen photo-sharing websites for measuring street attractiveness. Although the samples contained only images from London, it was believed that the discovered differences in photographic content would be also relevant for other urban areas. Indeed, as all three photo-sharing websites equally position themselves globally, the community of photographers can be expected to use them for similar purposes in every place where they are available.

A breakdown of mode classifications by image source revealed that sampled items from all three photographic datasets were real photographs with very few exceptions. This fact implied

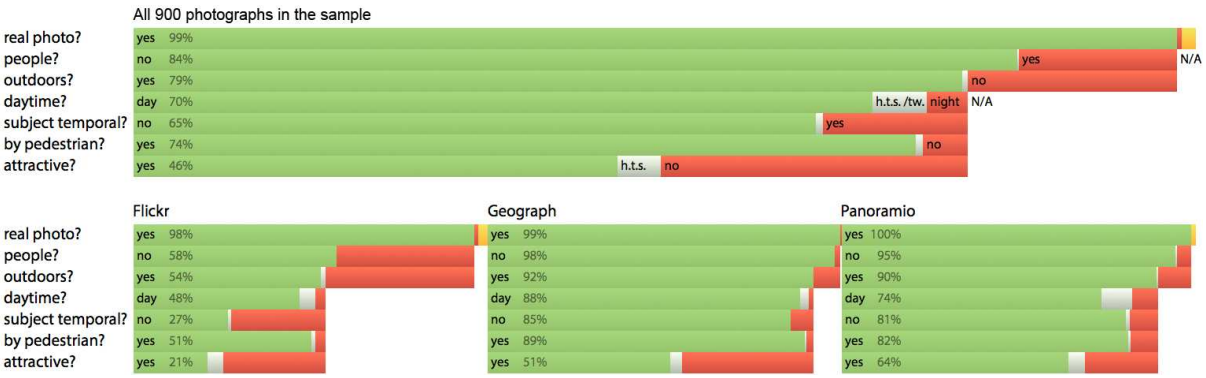


Figure 4.52: Aggregated results of the survey (mode answers). *Green:* suitable photographs in each classification, *grey:* hard to say, *red:* unsuitable photographs, *yellow:* photographs that were deleted at their origin since the launch of the survey and therefore could not be correctly assessed (displayed percentages are counted after their exclusion).
Data available at `http://github.com/kachkaev/survey-glyphs/tree/master/data`.

that requirement 1 to a *model photographic collection* (page 38) was satisfied in the vast majority of cases, and no filtering was needed in order to meet it.

Panoramio sample was found to contain the highest proportion of good quality 'votes', i.e. the photographs that depicted attractive walkable areas, according to the opinion of survey participants. Combined with a relatively clean initial spatiotemporal distribution, such property of image content made this photo-sharing service the closest to a *model photographic collection* prior to filtering.

Nearly a half of images in Flickr sample were classified as taken indoors, and about a half of outdoor photographs were marked as taken during events. As a consequence, the proportion of valid 'votes' for street attractiveness in this dataset was found very low (21%). Thus, without content-based and metadata-based filtering in addition to spatiotemporal filtering, Flickr could be hardly considered as a reliable source of street attractiveness scores. The scores, if calculated from the initial spatial distribution of photographs, were likely to be significantly overstated near popular buildings and common venues for events.

Although most of the images in the Geograph sample were passing all six formal requirements to the content of a suitable photograph, only a half of them were subjectively considered by survey participants as depicting attractive walkable places. This contradiction could be explained by the purpose of this photo-sharing service, classified as 'spatially explicit' by Antoniou, Morley and Haklay (2010). Geograph photographers are encouraged to fill in gaps in the spatial distribution of data rather than to share images solely based on personal preferences and past leisure walking experience. This makes community members deliberately capture underrepresented geographical locations, naturally increasing the proportion of invalid 'votes'.

Photo content assessment survey proved to be an important step in the analysis of candidate sources for street attractiveness scores. Subjective classifications of samples of photographs helped compare the datasets to each other and make conclusions about their overall suitability for the chosen purpose. Besides, the collected data enabled more analysis and a new set of filtering methods to be potentially proposed. These would be based on matching survey responses with the content of the photographs or their metadata.

4.5 Additional metadata and image content analysis

Having a combination of subjective classifications from survey participants together with the content of sampled images and attached metadata, it was possible to experiment with more potential filtering methods. Based solely on characteristics of individual photographs, these methods could reduce the proportions of irrelevant 'votes', making candidate datasets less distant from a *model photographic collection* and therefore help them become better estimators of street attractiveness.

This section describes the approaches that were attempted in this research project. Although not all of them were successful, the experience gained from the conducted experiments was found to be potentially useful when analysing other sources of crowd-sourced photographs or solve contiguous problems.

All experiments started with a proposal of a potential filtering method based on one or several available image parameters. The chosen features were loaded into the survey results interface (Figure 4.43 on page 177) and were represented in the list of glyphs, similarly to what is shown in Figure 4.51 on page 182. Then, the relationship between the given parameters and the responses from survey participants was manually explored by means of interactive visualization. If the examination of the data demonstrated potential, an approach to automated filtering for larger photographic collections was proposed, adjusted and verified. Finally, the new method was applied to the whole datasets, not only the samples.

In contrast to traditional statistical methods that could be blindly used to detect relationships between the available subjective and objective data, the chosen way of analysis enabled a deeper understanding of the contents of the given photographic collections and thus allowed more thoughtful decisions to be made.

The amount of computational effort required to perform metadata- and content-based filtering was significantly different, which was partially caused by a need to download and cache images in the second case (this issue is discussed on page 117). Therefore, it was found preferable to avoid the analysis of the bitmaps where possible and use them only when no alternatives could be applied.

4.5.1 Timestamp

Temporal photographic coordinates were widely studied at the stage of data distribution analysis (see Subsection 4.3.2 on page 147). It was then speculated that image filtering by a reported time of day in order to remove night photography (requirement 3.1.1 on page 38) would be inefficient due to possible frequent cases of incorrect camera settings. This statement could be confirmed or denied by looking at Flickr and Panoramio samples and comparing attached image timestamps to survey responses. Geograph could not be involved in the experiment because of absence of a 'time of day' component in the provided timestamps, as demonstrated in Figure 4.26b on page 147.

Because of different sunrise and sunset times thought the year, it was found reasonable to convert timestamps to a new derived variable – *the angle of the sun above the horizon*, which would reflect the actual time of day (Strous 2014). Ordering the list of assessed photographs by two human-assigned categories ('outdoors / indoors' and 'day / twilight / night', Figure 4.53)

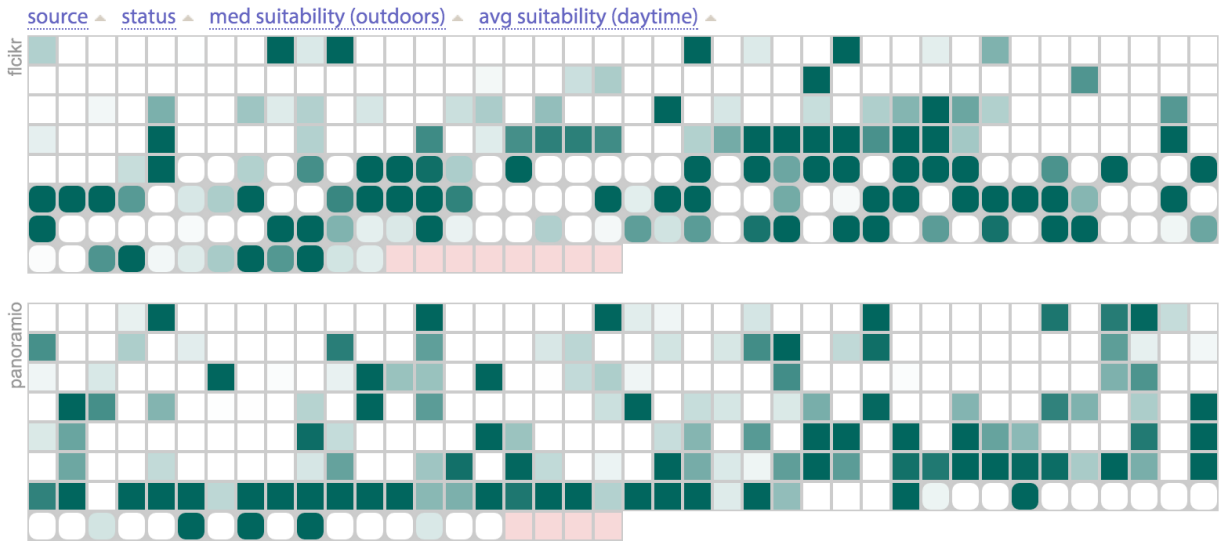


Figure 4.53: Row-prime ordered lists of assessed Flickr and Panoramio images coloured by time of day. White squares correspond to photographs reported as taken when the sun was 15° or higher above horizon. Dark aqua squares represent images taken when the sun was 5° below horizon or lower (also according to their time coordinate). Pale pink is used for images that were deleted at their origin after the launch of the survey, so had to be removed from the sample. Items with rounded corners are the photographs that were classified as taken indoors by the majority of respondents. Applied sorting mode is shown at the top of the figure.

while colouring items by the implicitly reported time of day confirmed the existence of errors in temporal coordinates. If photographs were filtered according to the angle of the sun above the horizon (or another similar derived variable, such as *minutes until the sunset*), quite a notable proportion of daytime images would be removed, especially in Panoramio. Given that the number of actual nighttime photographs in the random samples were rather small (Figure 4.52 on page 185) and because the proposed variable could not help remove indoor photography, it was once again chosen not to use time of day from timestamp for filtering. The same conclusion, however, does not necessary apply to *any* crowd-sourced photographic dataset or even to Flickr and Panoramio in the foreseeable future. With more network-enabled mobile devices at the photographers' disposal, the accuracy of the time of day in the attached metadata is eventually likely to improve. If time settings on a device are defined automatically in the majority of cases, timestamp becomes a good asset for distinguishing between daytime and nighttime 'votes'. Furthermore, in real-time image-sharing websites (such as photographic microblogs) missing or incorrect *time of shooting* can be potentially replaced with *time of posting* after confirming that the interval between these two events is small in most cases. The latter attribute can always be considered as reliable as it is assigned on the server side of a service.

4.5.2 Luminance from EXIF

One of the most significant discrepancies between image content in survey samples and what would be expected from a *model photographic collection* (page 38) was in presence of considerable proportions of indoor photography, especially in Flickr data (see Figure 4.52 on page 185). These images, certainly not being valid 'votes' for street attractiveness, could not only add noise to the resulting scores, but also introduce bias by considerably influencing road segments near popular buildings such as museums, shopping malls or restaurants. Hence, removal of indoor photography in the chosen datasets was regarded as an important task.

As indoor / outdoor image classification may be required for a wide variety of reasons, this problem was approached in many ways over the last 10–20 years (e.g. by Szummer and Picard 1998; Serrano, Savakis and Luo 2002; Pillai et al. 2011; Liu and Li 2013). Most of the discovered works focused on extracting of features from the bitmaps by means of edge detec-

tion, colour analysis and other applied methods of computer vision. Boutell and Luo (2004) proposed a combined approach for distinguishing indoor and outdoor photography by involving both image processing and the analysis of used camera settings. Among other things, this study showed that the attached "metadata cues alone can outperform content-based cues alone for certain applications, leading to a system with high performance, yet requiring very little computational overhead." The method relied on the values of shutter speed (exposure time), ISO speed, aperture (F number), subject distance and use of flash. These measures, combined with the features extracted from the images themselves, were loaded into a Bayesian network (Friedman, Geiger and Goldszmidt 1997), which was then trained on a sample.

Utilisation of camera settings attached to the photographs was mentioned in a number of other works, such as in a description of an image-annotating web service by Lee, Chen and Chang (2006) and in a U.S. a patent by Jain and Sinha (2008). The latter one comprises "clustering optical parameters of the digital images into a set of meaningful clusters, associating the set of meaningful clusters to a set of associated classes used by a user, and classifying the digital images according to the set of associated classes" (indoor / outdoor attribute is derived from a wider number of categories assigned to the photographs).

The success of scene classification solely based on metadata can be explained by the existence of a set of industry-wide standards for camera settings. The same scales are used by different manufacturers of photo equipment regardless of applied technical decisions for lenses or sensors. Thus, if one manually defines ISO speed, aperture and shutter speed equally on any of the two cameras and takes two photographs in the same environment, the brightness of the objects in the obtained images will be approximately the same. Similarly, if camera sets itself automatically based on the sensor-measured brightness of a scene, the settings will belong to a certain limited set of values, making it possible to apply them for classification.

Given that digital cameras use up to three degrees of freedom to achieve the correct brightness of the shots, the individual values of ISO speed, shutter speed and aperture may vary. However, their combination is always a derivative of the actual overall scene brightness to a high degree. This suggests that a correctly established function of camera settings can be used to 'reverse engineer' *the amount of light in the environment* where a photograph has been taken. Assuming that sunlight is stronger than artificial sources lights, this extracted measure *(luminance)* can

work as an estimator of a binary category of the scene – indoor photographs are likely to be associated with lower values of luminance compared to the daytime outdoor ones. The same applies to nighttime images, as they are taken under conditions of natural shortage of light.

Image classification into *daytime outdoors* versus *indoors or nighttime* based only on the extracted *value of luminance* was not expected to be as accurate as more comprehensive methods of analysis, especially those that involve consideration of photographic content in addition to metadata. However, if it could be shown that the approach was valid, it would become possible to significantly improve the quality of the photographic datasets in their ability to estimate street attractiveness scores with minimal added cost.

In order to know how to combine ISO speed, shutter speed and aperture into a value of luminance, it was decided to study the reverse process, i.e. how a camera is set up based on the amount of light in the environment. The theory described below originates from Jacobson (2000) and Lind (2000).

The process of camera setup starts with determining a required *exposure value EV* – the amount of light that needs to pass through the lens for a balanced photograph to be obtained. This number mainly depends on two factors: (1) the amount of light that a subject of a photograph is reflecting, or its *brightness B* and (2) the ability of a film or a digital sensor to absorb the light, or its *speed S*. These two factors form EV the following way:

$$\begin{aligned} EV &= SV + BV \\ SV &= log_2(0.32\,S) \\ BV &= log_2(B) \end{aligned} \tag{4.1}$$

BV and SV stand for *brightness value* and *speed value*. B is measured in *foot-Lamberts* and S – in *ISO ratings*. Logarithmic scale is used to ease manual calculations. BV is taken directly from a sensor called *luminance meter*, and the value of SV is known from a type of a film or from the analogous setting of a digital camera. Thus, obtaining EV becomes a matter of adding up these two values. Manufacturers of film and digital cameras tend to make the values of ISO speed unified and located at $1/3$ SV from each other (ISO 100, 125, 160, 200, ... 800, ... 3200, ... correspond to SV values of 5, $5\,1/3$, $5\,2/3$, 6, ... 8, ... 10, ...).

At the same time EV is a composite of two other parameters: *aperture* N and *shutter speed* t.

$$EV = AV + TV \tag{4.2}$$
$$AV = log_2(N^2)$$
$$TV = log_2(1/t)$$

AV and TV stand for *aperture value* and *time value*. Aperture is marked using f-number, corresponding to the focal length of the lens divided by the actual aperture. It is common to work with f/N notation instead of only N to show the relativity of this measure. The range of possible values is limited to a set of f-stops, usually located at $1/3$ AV from each other (e.g. f/N of $f/2.8$, $f/3.2$, $f/3.5$, $f/4$, ... $f/8$, ... $f/16$, ... correspond to AV values of 3, $3\,1/3$, $3\,2/3$, 4, ... 6, ... 8, ...). The bigger N, the smaller the aperture and the less amount of light goes through a lens at a unit of time. Shutter speed is measured in seconds and is also commonly limited to a range of fixed values with $1/3$ TV between each other (e.g. $1/30$, $1/40$, $1/50$, $1/60$, ... $1/250$, ... $1/1000$, ... give TV of approximately 5, $5\,1/3$, $5\,2/3$, 6, ... 8, ... 10, ...). The higher the shutter speed (i.e. the smaller the time t), the less amount of light is absorbed. The same captured brightness for a subject of a certain luminance can be achieved with use of equivalent pairs of N and t, e.g. $f/4$, $1/1000$ and $f/16$, $1/60$. The choice depends on the desired depth of field, motion blur, etc.

Equaling of the right sides of expressions 4.1 and 4.2 does not necessary guarantee the best possible light balance in a photograph – *exposure compensation* EC is occasionally introduced:

$$SV + BV - EC = AV + TV$$
$$log_2(0.32\,S) + log_2(B) - EC = log_2(N^2) + log_2(1/t)$$

EC of $\pm 1/3$, $2/3$, 1, etc. is needed in certain conditions to explicitly make captured objects look brighter or darker. This is done to make them more natural to a human eye or to achieve a special artistic effect. For example, when one takes a photograph of the mountains on a sunny winter day, EC of $+2/3$ or $+1$ helps avoid 'grey snow' (the value of EV is decreased, so either the aperture needs to be made bigger or the shutter speed has to be reduced; this increases the overall amount of incoming light). The value of EC rarely exceeds ± 1 or 2 EV and is usually equal to zero. Exposure compensation is referred as *exposure bias* in some literature.

Given the above relationship between the actual luminance of the captured objects and the camera settings, it is possible to 'reverse engineer' BV (or B) when all other parameters are known:

$$\begin{aligned} BV &= AV + TV - SV + EC \\ BV &= log_2(N^2) + log_2(1/t) - log_2(0.32\,S) + EC \end{aligned} \tag{4.3}$$

$$\begin{aligned} log_2(B) &= log_2(N^2) + log_2(1/t) - log_2(0.32\,S) + EC \\ B &= 2^{log_2(N^2) + log_2(1/t) + EC - log_2(0.32\,S)} \\ B &= \frac{N^2 \cdot 2^{EC}}{0.32\,t\,S} \end{aligned} \tag{4.4}$$

Restored luminance is able to give a prediction of the sought category of a photograph, however, this measure of light may contain hidden bias in variety of cases:

When a photographer attaches certain types of filters to a lens (e.g. a polariser), the amount of light that reflects from a subject and reaches a built-in luminance meter becomes lower. In these cases the actual value of BV may be greater than the extracted one.

Use of flash can result a substantial positive bias in BV, because the exposure is set up for this additional source of light rather than the actual conditions at the scene.

Luminance can slightly vary for two objects photographed under the same conditions if the reflective properties of the materials they are made of are too distinct (this value of luminance should not be confused with the *illuminance* of sources of light at the scene, which would be a better predictor of the type of the environment).

Professional photographers may explicitly overexposure or underexposure their works and then correct the brightness of objects in a graphics editor. This can be done to avoid 'clipping' – a situation when certain areas of an image become 'pure white' or 'pure black' as the intensity of light goes beyond the maximum or the minimum of what can be captured by a digital sensor.

The above conditions except for the use of flash cannot be known from image metadata, which makes luminance a noisy characteristic to some degree. However, the amount of this noise in most of the cases cannot be enough for a photograph to be misclassified.

Later in this work the use of term *luminance* refers to BV, a logarithm of the subject brightness B. It is measured in *steps*, or $log_2(foot\text{-}Lambert)$.

Camera settings, which were necessary to estimate the values of subject luminance in the chosen data sources, could be harvested from Flickr, Panoramio and Picasa. The required attributes were accessible because these web services read and stored EXIF tags of the original uploaded photographs (Camera & Imaging Products Association 2010), as shown in the examples in Figure 4.54. An experiment on the luminance-based image classification was not possible for Geograph due to data unavailability. It was also cancelled for Picasa, because this photo-sharing service was rejected at the stage of data gathering (Section 4.2).

Raw EXIF attributes were first collected for those Flickr and Panoramio photographs that were assessed by survey participants. The values of ISO speed, shutter speed, aperture and flash usage were parsed and manually verified. The second step was necessary due to a discovered inconsistency in the used data formats. For example, the value of aperture could be located in tags '`ExifIFD.Aperture`', '`EXIF.Aperture`', '`IDF0.Aperture`', '`TIFF.Aperture`' and was presented as '`50/100`', '`2`' or '`f/2.000`'. The number of possible values for flash status was over fifty (including both numeric and textual encoding).

Camera	Nikon D5100
Exposure	0.05 sec (1/20)
Aperture	f/4.8
Focal Length	45 mm
ISO Speed	2500
Exposure Bias	+1/3 EV
Flash	No Flash

Camera	Nikon D3000
Exposure	0.033 sec (1/30)
Aperture	f/3.5
Focal Length	18 mm
ISO Speed	320
Exposure Bias	0 EV
Flash	Off, Did not fire

Camera	Canon EOS 400D Digital
Exposure	0.01 sec (1/100)
Aperture	f/11.0
Focal Length	18 mm
ISO Speed	100
Exposure Bias	0 EV
Flash	Off, Did not fire

Figure 4.54: Examples of EXIF tags with recorded camera settings. Photographs are courtesy of their authors and are shared under *Creative-commons* licences.

Calculated luminance and flash usage values were loaded into the survey results analysis tool and were explored both on a scale of the whole sample and by individual images. Different visual representations of the data, such as one in Figure 4.55, confirmed that these derived attributes had a potential for excluding invalid 'votes' for street attractiveness in crowd-sourced photographic collections. The majority of daytime outdoor images were found having a considerably higher luminance than those taken indoors or at night. Besides, flash was mostly used in the second category of cases, which made it a useful additional input for filtering.

Few occurrences of daytime photographs with a relatively low luminance or with flash were images taken on a cloudy day or right before the dusk. Indoor pictures with no use of flash but still with a high luminance were mostly those that were taken in atriums (parts of shopping malls, museums or railway stations that have glass roofs).

Interactive visualization showed that processed EXIF attributes made it possible to significantly improve the quality of an average 'vote' for street attractiveness in both Flickr and Panoramio datasets. The following simple rule could be applied: 'all photographs with luminance lower than a certain threshold or those taken with flash should be rejected'.

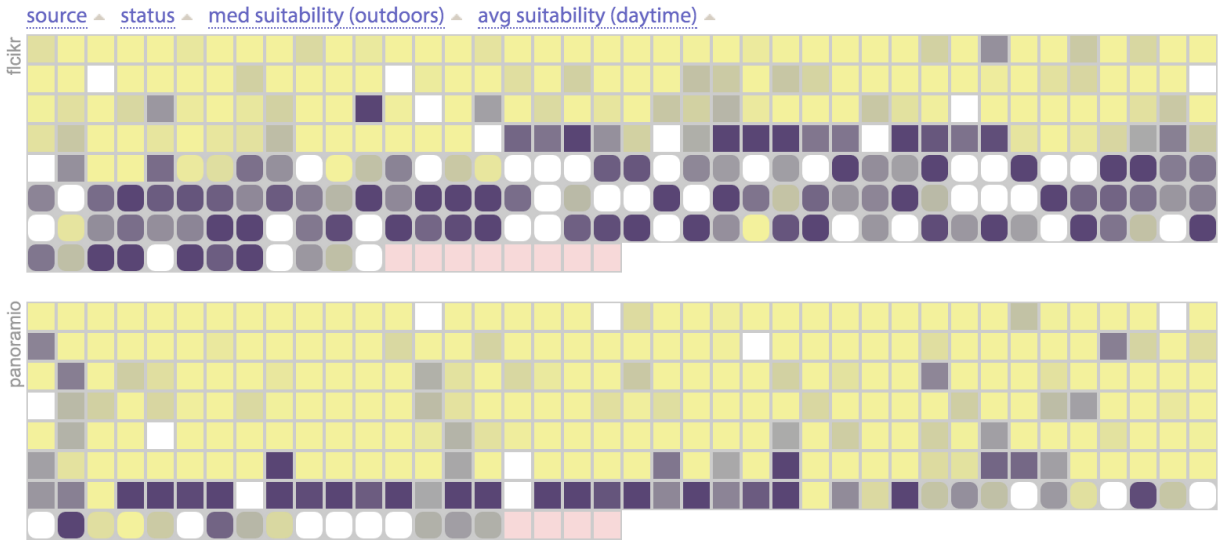


Figure 4.55: Row-prime ordered lists of assessed Flickr and Panoramio images coloured by extracted luminance. Bright yellow squares correspond to images with higher values of luminance (>10), and the more towards grey (≈3–4) and then dark purple (<0), the smaller the value. White squares correspond to cases when flash was fired. Pale pink is used for images that were deleted at their origin after the launch of the survey, so had to be removed from the sample. Items with rounded corners are the photographs that were classified as taken indoors by the majority of respondents. Applied sorting mode is shown at the top of the figure.

The higher the threshold was to be chosen, the bigger proportion of indoor and nighttime photographs could be removed, but the more cases of inexpedient filtering would take place. On the other hand, excessive lowering of the threshold could make filtering less efficient as more irrelevant 'votes' would remain in the data.

An optimal value for the luminance threshold was chosen by means of a short statistical study, which consisted of the following steps. First, an evenly distributed set of candidate thresholds was taken from a range of obtained luminance values; the size of a step between the candidates was made equally small (0.25 of a step). Then, the photographs were classified into 'passed' and 'rejected' for every given threshold according to the chosen rule. Next, the results were matched against the aggregated subjective survey classifications, which played a role of the 'ground truth'. Finally, a number of statistical measures such Chi Square and percentages of errors were used to represent adequacy of filtering in each case. The details of the study can be found in Table C.1 on page 294 (Appendix C), and a summary of its results is shown in Figure 4.56.

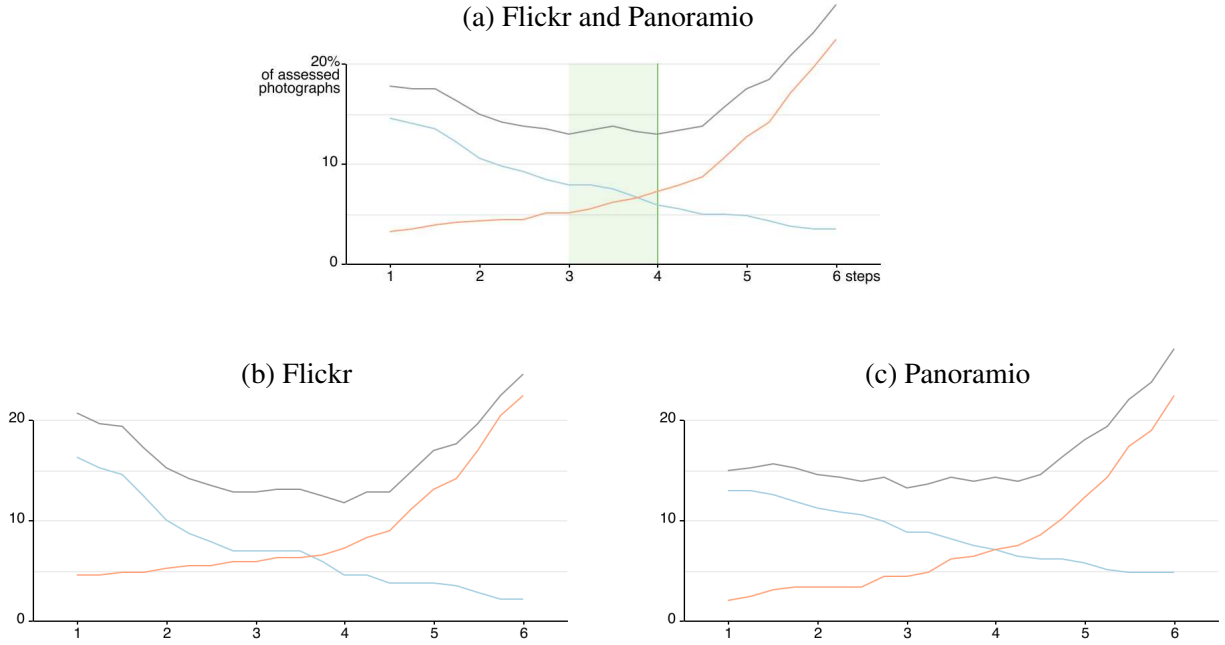


Figure 4.56: Comparison of different values for luminance threshold with step of 0.25. The charts show frequencies of false positives (cases when survey participants marked a photograph as taken at night or indoors, but is was classified as daytime outdoors), false negatives (cases when survey participants marked a photograph as taken outdoors during daytime, but but based on the chosen luminance threshold is was classified as taken indoors or at night) and their sum.

With statistical analysis, it was possible to verify the overall reliability of the proposed filtering method and reveal that the value of luminance between 3 and 4 steps would be optimal. It was chosen to use the rightmost boundary of this interval, as this was making filtering more 'strict', potentially leaving less biased irrelevant 'votes' in image distributions. When during one of the experiments flash status was ignored and the photographs were classified only by their luminance, the same value of threshold remained optimal, but the numbers of errors slightly increased.

After this successfully conducted trial, luminance and flash status were extracted for those Flickr and Panoramio photographs that were not sampled for a survey. A distribution of these features in the latest versions of the datasets are shown in Figures 4.57.

The forms of the bar charts showing all obtained values agreed to the findings of the survey and the statements made earlier in this subsection. There appeared to be more images with a lower luminance in Flickr than in Panoramio, corresponding to a higher proportion of indoor photography and pictures taken at night. Use of flash was relatively more often for intermediate

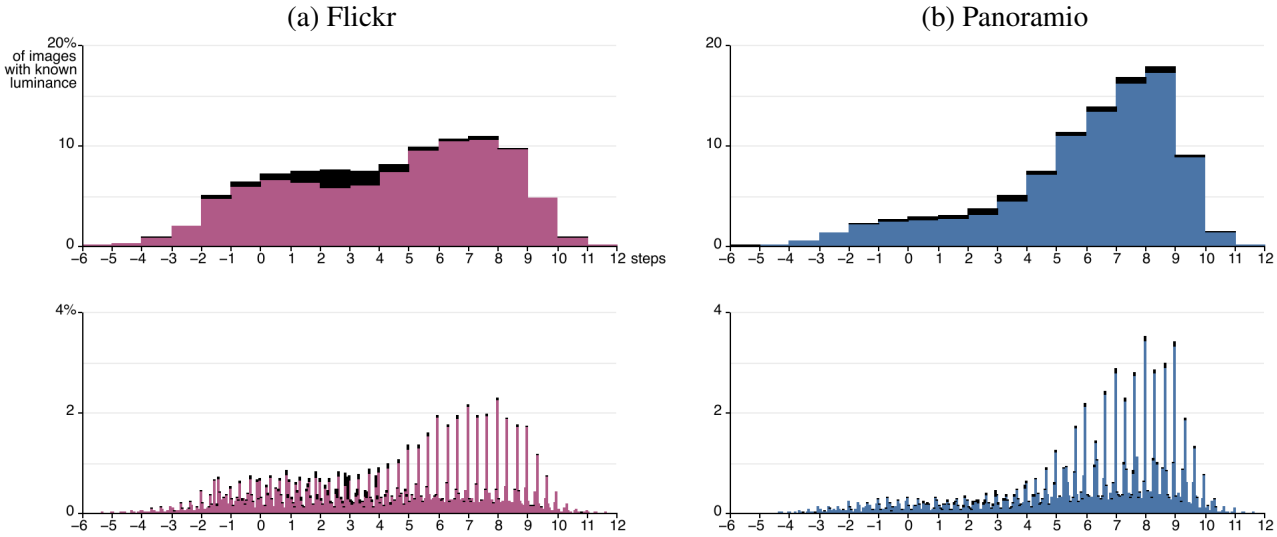


Figure 4.57: Distributions of extracted luminance in the latest versions of the datasets with bins of 1 step *(top)* and $1/18$ steps *(bottom)*. Black fragments of bars correspond to cases when flash was fired. All four stacked bar charts are formed only by photographs reported as taken between 2008 and 2013; images with undetermined luminance are not included; no other filtering of data has been applied.

values of luminance – these items were unlikely to be daytime outdoor photography and would appear closer to the left-hand side of the charts if flash was disabled. Observed local peaks at $1/3$ of a step were in line with the approach that cameras use for exposure setting – SV, AV, TV and EC are commonly defined with exactly this precision. All 'reverse-engineered' values were found within the measuring limits of luminance meters available on the market (for example, a professional general-purpose luminance meter Konica LS-100 (Konica Minolta 2013) detects B from 0.001 to 87,530 *foot-Lamberts*, which corresponds to BV of -11.5 to 14.8 steps).

The benefit from image filtering by EXIF luminance could be clearly seen when photographic records were spatially arranged and either grouped or coloured, as shown in Figures 4.58 and 4.59 on two following pages. In both Flickr and Panoramio datasets the images with lower luminance and enabled flash were found mainly concentrated around popular public buildings such as museums, railway stations, pubs, etc. Without their detection and removal, these potential 'votes' could significantly bias attractiveness scores and consequently mislead the routing algorithm.

EXIF attributes that were required for obtaining luminance were found unavailable or defective for some of the images – about a quarter of those in the initial Flickr dataset and almost one out of ten in the initial Panoramio dataset (the difference between these numbers was partially caused by an earlier rejection of a bigger proportion of Panoramio images due to absence of time of photographing, also derived from EXIF). The attributes could be missing because they were not originally uploaded to the photo-sharing websites or were made unavailable by their authors in privacy preferences. In a number of cases only particular tags were lacking, e.g. everything except ISO speed or shutter was specified. The attributes for few photographs were found defective, i.e. had no semantic meaning (for example, EC value for one Panoramio photograph was equal to $1,431,655,808.00$ steps, suggesting that there occurred a problem related to storing data in a binary format).

Default action for 'votes' that miss the value of luminance may depend on the overall expected proportion of indoor and nighttime pictures in a given photographic source. If the vast majority of the content is daytime and outdoors, these records can be treated as 'passed'. Although such strategy may add bias to the final spatial distribution of the 'votes', it can protect the dataset

Figure 4.58: Spatial distribution of extracted EXIF luminance in the latest version of Flickr dataset. The maps are formed by photographs reported as taken between 2008 and 2013; no other filtering of the gathered data has been applied before producing the images. The distribution of photographs with the known luminance and no flash *(middle)* is split into six small multiples *below*, each corresponding to a certain value (and colour) range.

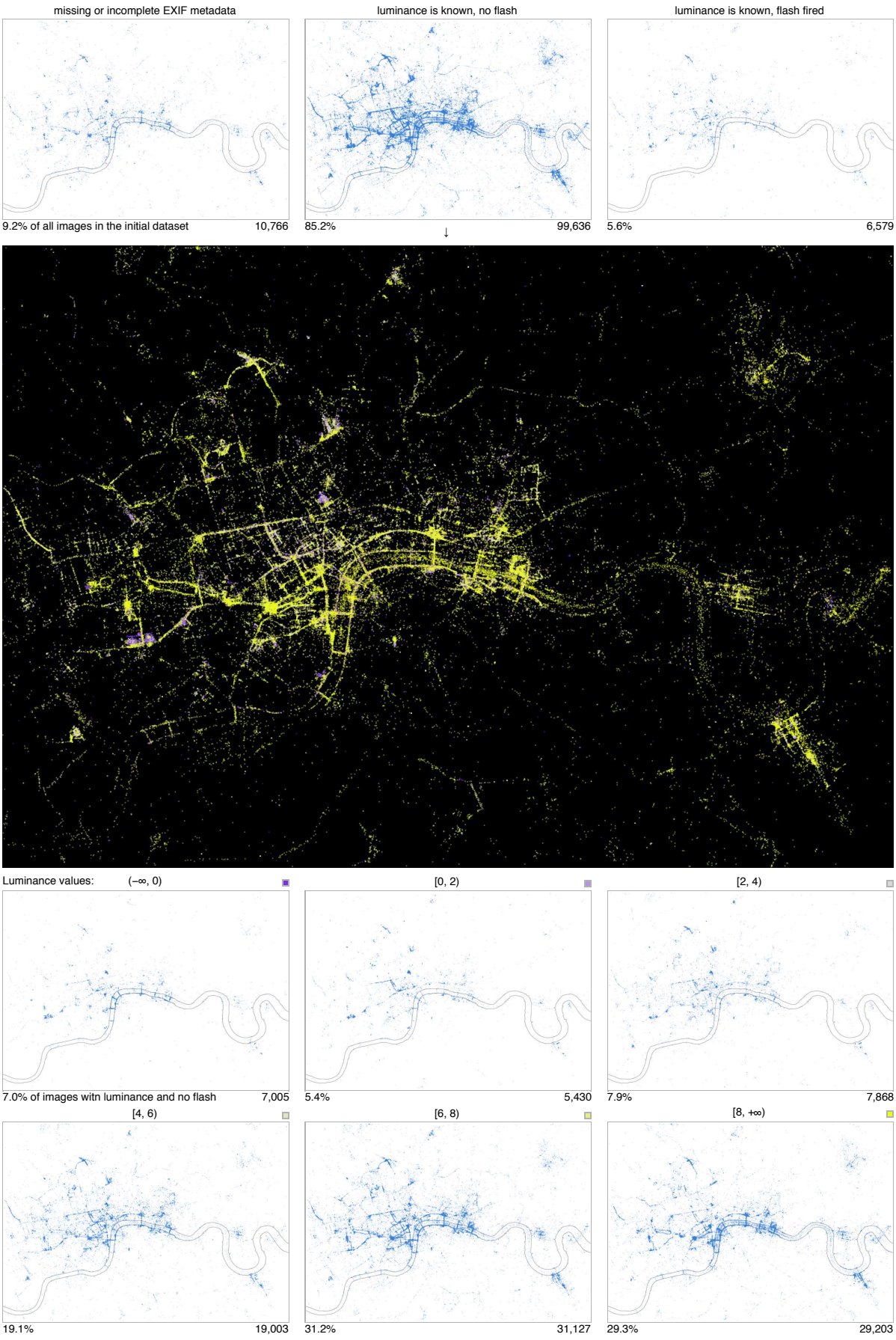


Figure 4.59: Spatial distribution of extracted EXIF luminance in the latest version of Panoramio dataset. The maps are formed by photographs reported as taken between 2008 and 2013; no other filtering of the gathered data has been applied before producing the images. The distribution of photographs with the known luminance and no flash *(middle)* is split into six small multiples *below*, each corresponding to a certain value (and colour) range.

from becoming too small and thus too noisy. In this work, both Flickr and Panoramio images with unknown luminance were marked as 'rejected' for consistency despite that a different default action could be used in the second case.

Taking into account the distribution-based filtering methods (Section 4.3), the number of 'votes' to be excluded due to unknown luminance can be reduced. Filtering functions that precede the extraction of luminance and imply 'randomness' in decision-making can be made informed by EXIF tags too and give preference to the photographs that have this information. A disadvantage of this improvement is in a need to download metadata for a significantly bigger number of records, which may result into longer interactions with photo service APIs. This workaround was used in this project, because metadata for all images had to be downloaded for research purposes anyway.

4.5.3 Textual attributes

Users of photo-sharing services often supplement their uploads with title, description and tags, which characterise the content of the images. These textual attributes serve both personal and social purposes, for example allow photographers to organise their own memories or to receive extra attention from other community members (Ames and Naaman 2007). The quality of textual information differs from person to person and from photograph to photograph – some annotations are rather explicit, while others are brief or empty (Mamei, Rosi and Zambonelli 2010). Variations in data also exist on a macro level – the proportion of completed textual fields appear to be smaller in the gathered Flickr and Picasa datasets compared to those from Panoramio and Geograph.

A large number of previous studies focused on textural attributes in crowd-sourced photographic data (e.g. Dykes et al. 2008; Kennedy and Naaman 2008; Crandall et al. 2009; Lee, Greene and Cunningham 2011; Popescu and Grefenstette 2011). They demonstrated the applicability of these attributes in a wide range of spatially-oriented tasks, such as for describing geographic regions or detecting popular landmarks. Textual data was also utilised in a few photo-based trip planning projects, overviewed in Chapter 2. For example, Popescu and

Grefenstette (2011) suggested landmarks to users of photographic services after comparing their own tags with tags in images by other users. Kurashima et al. (2012) used textual attributes to respond to interests of people who were planning a trip.

In the context of this research, title, description and tags could inform binary filters in two ways. First, there could be created a list of *approved terms*, allowing only items annotated with these terms appear in a final distribution of 'votes' for street attractiveness. Second, there could made a list of *stop-words*, presence of which would mark photographs as rejected. Any of the two lists could be formed based on the results of previous research and by conducting a statistical study.

Both potential scenarios incorporated several fundamental and project-specific issues, which could not be bypassed. First of all, filtering could not be applied equally to all photographs due to briefness or absence of textual information in many cases. Some titles or descriptions were automatically generated, especially in Flickr data (e.g. '`Shot 142`' or '`DSC00184`'). Second, a properly defined list of filtering terms had to be translated to all popular world languages to avoid language-driven bias and to make the same method applicable in other cities. Hundreds of annotations in Arabic, Chinese, Cyrillic or Greek alphabets were observed even within London area during a manual exploration of Flickr and Panoramio datasets. In addition to support of multiple languages, filtering had to be made robust to typos, which would make the implementation more complicated. Finally, introduced stop-words or approved terms had to be tested on large datasets with a known 'ground truth'. With only 300 manually assessed images per dataset it was impossible to estimate the quality of text-based filtering due to a much higher number of possible unique cases compared to the sizes of samples. Some work in this direction was recently done by Quercia, Schifanella and Aiello (2014).

Summarizing the above, it was decided not to introduce text-based image filtering in this research and also not to recommend this approach in cases when the photographic 'votes' for street attractiveness have binary, not linear weights. These attributes, however, still have a potential to inform path selection in a leisure routing system that implements fuzzy logic for image filtering (this concept is briefly described on page 3.1.1).

4.5.4 Moderation category

All geotagged images in Geograph and Panoramio are moderated by members of staff shortly after they are shared by photographers. The outcome of moderation is publicly available together with other metadata, which makes this attribute potentially useful for excluding irrelevant 'votes' for street attractiveness from these crowd-sourced collections.

Moderators at Panoramio assign recently uploaded photographs with one of the following three categories: 'selected for Google Earth', 'selected for Google Maps and Google Earth' and 'not selected for Google Earth or Google Maps' (Panoramio 2014). The first category if very similar to the second one is almost never used. In order for a photograph to be 'selected', it needs to be in line with two acceptance policies listed on the website. The first policy is rather general and is needed to protect visitors from content that is totally unacceptable (e.g. photographs of nudity or violence and images that violate the copyright law). Such items are permanently deleted once seen by a moderator. The second policy distinguishes the photographs that *describe the surrounding environment* from the rest of the accepted contributions. The criteria used by moderators here are very similar to the requirements suggested for a *model photographic collection* on page 3.1.1. Both 'selected' and 'not selected' photographs can be retrieved with spatial search and are also accessible via individual URLs.

According to the documentation on Geograph website, a photograph can belong to one of four categories assigned by a moderator. These are 'rejected', 'pending', 'accepted' and 'geograph'. Only photographs marked as 'accepted' and 'geograph' appear in the publicly available database dumps (Geograph 2014*a*). The concept of 'geograph' category is similar to what is meant by 'selected' in Panoramio, however, a set of underlying rules is more strict (Geograph 2014*b*). For example, a 'geograph' image must have a textual description and "clearly show one of the main geographical features within the square" (i.e. a cell of the Ordinance Survey National Grid to which a geotag belongs). "Pictures of small features (e.g. mileposts, telephone boxes), parts of buildings, flowers, butterflies, etc." are classified as 'accepted', not 'geograph' as well.

In order to understand if moderation category may or may not be involved in binary filtering of 'votes' for street attractiveness, this attribute was loaded into the survey results analysis tool, as demonstrated in Figure 4.60 on the next page.

A relationship between the categories assigned by moderators and the subjective responses left by survey participants could be easily studied with interactive visualization. The interface (Figure 4.43 on page 177) made it possible to see the overall proportions of 'approved' images and also investigate why each individual item was placed to one of the two categories.

It was confirmed that moderation in Geograph was considerably more 'strict' than in Panoramio, resulting 69% of 'approved' photographs versus 87% in the second dataset. In both cases there existed a correlation between how suitable the images were found by respondents and what moderation category was assigned. None of the indoor Geograph photographs were approved by the moderators compared to 10 out of 26 in Panoramio. Besides, there existed significantly more relevant 'votes' that were not approved, including those that passed all requirements to the content in a *model photographic collection* (page 38). None of such cases were found in Panoramio during a detailed exploration of the assessed sample of images.

Given that the spatial density of the initial Geograph collection was small, and filtering by moderation category could reduce it further by about 30% while offering only little improvement, it was decided not to apply filtering for this source of images. Panoramio data suggested an opposite decision – moderation-based filtering was found a useful addition to a luminance-based approach for exclusion of some types of irrelevant 'votes'.

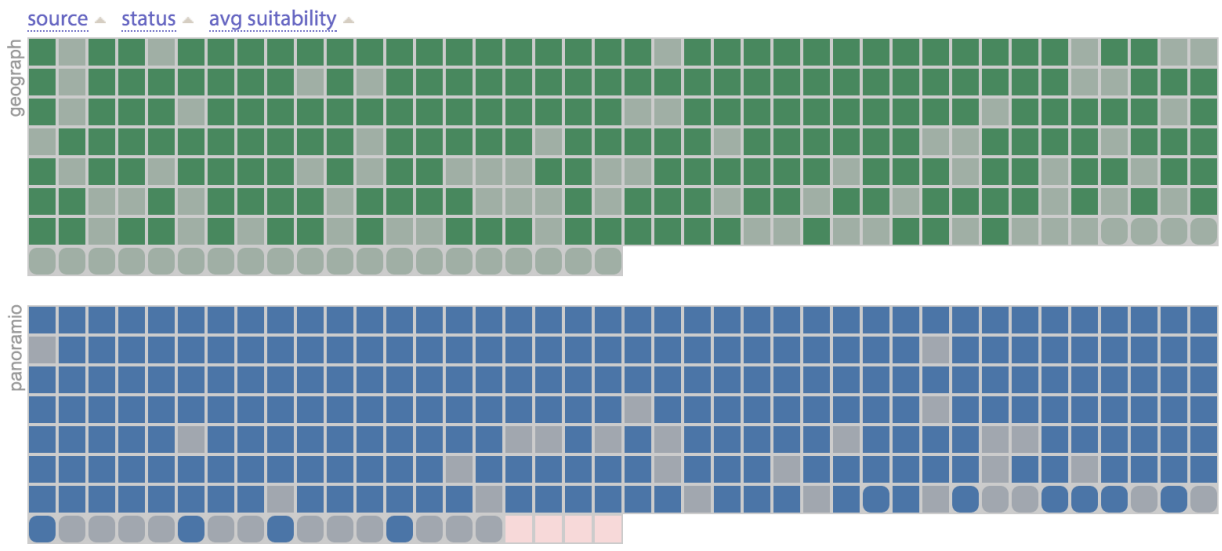


Figure 4.60: Row-prime ordered lists of assessed Flickr and Panoramio images coloured by moderation categories. Geograph photographs are either marked as 'geograph' or 'accepted'. Images from Panoramio are either 'selected' or 'not selected' for Google Earth and Google Maps. Pale pink is used for images that were deleted at their origin after the launch of the survey, so had to be removed from the sample. Items with rounded corners are the photographs that were classified as taken indoors by the majority of respondents. Applied sorting mode is shown at the top of the figure.

4.5.5 Face detection

The survey revealed that the main objects in a proportion of assessed photographs were humans rather than the elements of local urban environment. This suggested that the removal of such instances could potentially help photographic datasets produce less noisy distributions of 'votes' for street attractiveness. The issue of presence of people on the photographs was mostly relevant to Flickr – about 42% of sampled images from this source were marked by respondents as featuring one or more individuals.

It was suggested that an arbitrary crowd-sourced photographic collection could be divided into subsets of 'human-centred' and 'environment-centred' images by mining sizes and locations of faces in all photographs. These features could be either obtained as a pre-determined array of polygons provided by a photo-service API (if possible) or extracted explicitly with use of an automatic face detection algorithm. The first approach would be less cost-effective, but could be only applied when faces were accurately annotated either by the users of a photo-sharing website or automatically by the server. The second scenario was more complicated – it implied caching image content and running a face detection tool to obtain the result.

Among the three data sources that were included in the survey, externally determined face annotations were only available for Flickr. Regions with people in the photographs on this photo-sharing service were marked with rectangles and could be obtained by calling a special API method, querying one image at a time. Available metadata, however, was found far from reliable – the annotations existed only in a very small number of cases. The reason for such a poor quality of the face rectangles was that they could be only defined manually, and this additional action did not appear to be a habit of the majority of the users.

To know if exclusion of photographs with *automatically detected* faces could make the collections of 'votes' for street attractiveness more adequate, a short empirical study was organised. First, the faces in all nine hundred photographs, which were chosen for the survey, were manually annotated. This was done with the same interface that was used to examine survey responses (see Figure 4.43 on page 177 and the left-hand side of Figure 4.61 on page 208). Second, the results of annotation were loaded into new custom designed glyphs, compactly showing all added face rectangles (see the top part of Figure 4.62 on page 209). A total of 248 faces

in 124 photographs were found. Finally, with various reorderings of the list, details on demand and other interaction methods, a number of preliminary conclusions were made: (1) It was confirmed that *presence of faces* was in a very strong correspondence to *presence of people* in the photographs, according to survey responses. Few number of false positives included sculptures, advertisements and small-scale artworks. (2) A proportion of photographs with faces in Flickr sample was significantly higher than in Geograph and Panoramio. (3) Presence of faces was mostly peculiar to the images that were not classified as 'suitable' by survey participants. The above findings suggested a more detailed investigation of how presence of human faces could be involved in binary filtering of photographic datasets. To know if automatic face detection could be feasible and reliable for the task, another study had to be carried out.

The outcome of filtering by automatically detected faces would depend of three parameters: configuration of an involved face detection algorithm, resolution of images to process and approach to result interpretation. The first parameter was the most crucial to the success of the method. As the quality of face detection varies in different face detection applications (Degtyarev and Seredin 2010), it was important to use the best available option. The bigger the resolution of cached images, the less number of errors were likely to occur, but the more resources were needed. Hence, it was necessary to find a balance between the performance of the process and the quality of the result.

The problem of automatic face detection and face recognition in still photography or in video streams has been widely research in the recent years (Grgic and Delac 2007; Mashape 2013). Some of the existing technologies are able to give extremely accurate results, however, not all of them are available free of charge. It this research it was decided to focus only on two commonly applied face detection libraries, but run the experiments using several available configurations, thus increasing the number of tests.

The first chosen library, OpenCV, is a general-purpose computer vision tool, which uses Haar Feature-based Cascade Classifiers (Viola and Jones 2001). In order for face detection to work, the algorithm needs to be trained on a sample of photographs, which are manually prepared beforehand (OpenCV 2014). Once this process is complete, the library can search for previously seen patterns in an arbitrary collection of images and return the locations of familiar objects within them. The accuracy of the method vastly depends on a training set that

has been used. For example, if it consists only of frontal projections of faces, side views are very unlikely to be identified. OpenCV provides a list of pre-trained and tested Haar cascades (`https://github.com/Itseez/opencv/tree/master/data/haarcascades`), which can be used by the public for various tasks, including face detection. Five of these cascades were selected for the experiments: `front`, `frontalt`, `frontalt2`, `frontalttree`, `profile`.

The second chosen library, Core Image, is an image processing and analysis technology maintained by Apple. It is commonly used in native OSX and iOS applications for a wide range of tasks, including face detection (Apple 2013). The principle of the algorithm is not disclosed in the official documentation. Core Image face detector can work in two modes: with `low` and `high` accuracy. The second one is described as more resource intensive, but potentially leading to less false positives and false negatives. It was chosen to involve both modes in this research. Because Core Image face detector could be only used inside programs written in Objective C, a command-line wrapper was implemented to make its functionality available from the Dataset Abstraction Framework. The source code of this tool is available at `https://github.com/kachkaev/CICommandLineFaceDetector`.

Photo-sharing services convert the original uploads into files with different sizes, each available via a separate URL. This makes it possible to run face detection with a varying balance between the quality and the cost. The number of correctly located faces can be expected be higher for larger images, but it is also likely that marginal expenses can become higher than marginal gain at some point. Choosing a right compromise is extremely important when automated face detection needs to be done for a large remote collection of photographs. Given the list of available image sizes for all three assessed data sources (Table 4.2 on page 118), it was chosen to run the experiment for three of them, that is 240, 500 and 1024 pixels on the longest side. Due to a slightly different standard used at Geograph, a fallback of 213, 640 and 640 pixels was introduced for this source (one image size was used in two out of three groups of experiments). If some of the assessed Panoramio photographs were not available as 1024-pixel files, a fallback of 640 pixels was used as well.

Because there was no proved evidence that faces of any size were making a photograph ‘human-centred’, it was decided to check if introduction of some threshold could potentially lead to an improvement in the outcome of filtering. In fact, small face rectangles (e.g. with

maximum edge size less than 5% of the longest side of a photograph) could potentially belong mostly to passers-by, making the exclusion of such cases not necessary. In this research it was chosen to use thresholds of 0, 5, 10, 20 and 30% (detected faces smaller than this proportion of the longest side of an image should not make it 'human-centred', i.e. rejected).

Thus, the goal of the experiment was to see if any combination of the above three parameters could help reliably distinguish between the valid 'votes' for street attractiveness and the photographs that needed to be rejected. In case of a positive outcome, the most optimal configuration could be used for classifying larger datasets.

Automatic face detection was run for every photograph in a survey sample 21 times, that is seven algorithm configurations by three image sizes (the photographs were cached beforehand as explained in Subsection 4.2.1 on page 111). The process took 46 minutes for large images, 21 minutes for those of middle size and 9 minutes for the smallest ones. The results were visualized in survey analysis tool, as shown in examples in Figure 4.61 below and Figure 4.62 on the next page. Simple interactive exploration of data revealed significant numbers of false positives in all five configurations of OpenCV classifier and demonstrated a higher general reliability of Core Image. Haar cascades were detecting non-existing faces in a great variety of circumstances, however, were rarely useful in cases when real faces were photographed from a non-standard angle or appeared under unusual lighting conditions. Both versions of Core Image gave considerably less false positives, but still were not always correct.

Figure 4.61: Face detection examples. *Left*: the process of manual face annotation. *Middle* and *right*: successful and unsuccessful automatic face detection. The frames are coloured by the type and the configuration of the used face detection algorithm configuration. Photographs are courtesy of their authors and are shared under *Creative-commons* licences. Faces were blurred for the purpose of confidentiality.

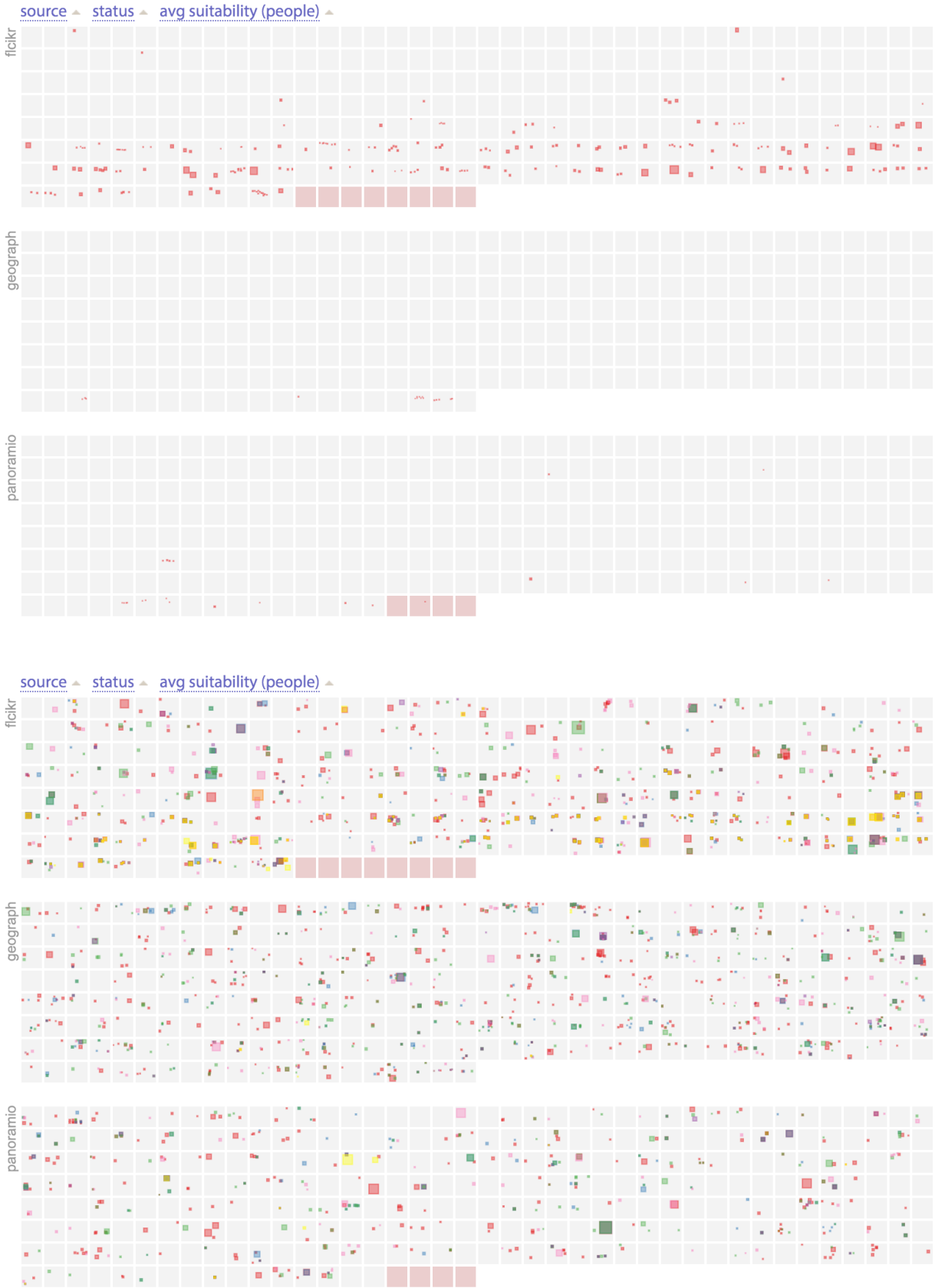


Figure 4.62: Row-prime ordered lists of assessed Flickr, Geograph and Panoramio images with positions of human faces. *Top:* manually marked faces. *Bottom:* Automatically detected faces in photographs of medium size. Colours correspond to different face-detection libraries and their configuration (see page 207). Pale pink fill is used for images that were deleted at their origin after the launch of the survey, so had to be removed from the sample.

Visual analytic approach was not able to lead to the main outcome of the experiment, so it was necessary to support the observed findings with statistical evidence. First, automatically detected faces were compared to the manually annotated ones, which in the given context were considered as ground truth. The results of this comparison are shown in Table C.2 on pages 295 and 296 (Appendix C). Measured numbers of positives, negatives, false positives and false negatives revealed the quality of face detection in every combination of parameters. For the cases when the value of a face size threshold was not zero, a 'positive' was a situation when both manually annotated and automatically detected faces were bigger than the minimum defined relative size. These derived variables were also supplemented with a calculated Euclidean sums of squares of the latter two measures.

The statistics confirmed that Core Image face detectors were more reliable among the two considered algorithms, but no significant benefit of using high accuracy mode in favour of a standard one was revealed. As expected, the difference between automatic and manual face detection was the smallest when processed bitmaps were of the largest size, however, a gap in the outcome for images of 500 and 1024 pixels long was not substantial. Numbers of false positives and false negatives were naturally smaller for higher face size thresholds, as face detectors were better at dealing with larger features.

The fact that Core Image in low accuracy mode could potentially give a reasonably good prediction of the presence of human faces in crowd-sourced datasets did not necessary imply the usefulness of 'vote' filtering by this extracted feature. Removal of photographs with faces was only reasonable if (1) their presence was a good sign of 'non-random' people in the photographs and (2) if photographs with people as main objects were likely to be irrelevant 'votes' for street attractiveness. To test these two relationships, manually annotated and automatically extracted faces were matched against mode answers from survey participants using a chi-square test. The results are provided in Table C.3 on page 297 (Appendix C). This analysis confirmed both of above hypotheses for some combinations of image size and face size threshold. Based on these data and data on algorithm performance rates it was found optimal to run face detection on images of the medium size and not to set any face threshold.

In this research face-based filtering was only applied to Flickr photographs, because of very too few occurrences of faces in two other datasets. Apart from that, it was explored that in a sample of 'human-centred' Panoramio images all were marked by service moderators as 'not selected', which also suggested that no further data cleaning was needed.

4.5.6 Greenness of space

Another idea for removing irrelevant 'votes' for street attractiveness from crowd-sourced photographic data was based on previous research of urban vegetation. Some studies (e.g. Greater London Authority and London Development Agency 2003; Sugiyama et al. 2008) have demonstrated that green areas make a positive impact on many aspects of human life from physical and mental health to real estate prices; they also increase walkability of the surrounding areas. The fact that greenness of space is widely appreciated by pedestrians could be potentially utilised in a photo-based leisure routing system. If it could be shown that the images with captured vegetation were significantly more likely to suggest a good place for a walk, it would become reasonable to clean the distributions of 'votes' for street attractiveness from photographs with no or little green regions.

It was decided to test the above hypothesis using a similar approach to the one applied for face extraction. The first step was to manually annotate the assessed photographs with some measure of greenness, which would represent a proportion of pixels having a certain range of hue. Then, the obtained scores were to be visualized in the survey analysis tool and explored. If, like in the case with human faces, the exploration demonstrated some vivid relationship between the amount of green in the images and the subjective opinion of survey respondents, further steps would include automated estimation of green and a statistical study.

Instead of manually outlining all noticed green areas using polygons (which could take more time than annotating faces with rectangles), sampled photographs were assigned with a numeric score, representing *an approximate proportion* of image pixels with green hue. The values for this score were defined as following:

0, if a photograph contained no observable areas with green hue;

1, if there were minor occurrences of green, which were not instantly noticeable;

2, if a bitmap had one or several patches with green hue, yet these patches were not dominating in the image;

3, if more than about 40 – 50% of pixels were shades of green.

Automatic classification of photographs into those that contain and do not contain vegetation can be problematic in some crowd-sourced datasets, as green hue may also correspond to artificial objects. If 'man-made' green image regions are not a rare phenomenon, a simple summation of pixels of a certain hue becomes not enough for detecting 'natural' greenness. In order to explore the scale of this potential issue in the given samples, manually assigned scores were multiplied by -1 when there were more artificial green areas than the those of vegetation. Some examples of how the images were classified can be found in Figure 4.63.

Distributions of manually assigned greenness scores in the assessed photographs are shown in Table 4.3 and in Figure 4.64 on the facing page. Despite that the chosen method for annotation was not robust, and the results could contain perceptual errors or bias, it still allowed for some general conclusions about the data to be made. Interactive exploratory search for any relationships between obtained greenness scores and survey responses did not reveal any patterns that could potentially be significant. Thus, it was concluded that no binary filtering by 'image greenness' would be useful for the chosen task. According to the opinion of survey respondents, images featuring various amounts of vegetation were almost as likely to be irrelevant 'votes' for street attractiveness as photographs with no green areas at all.

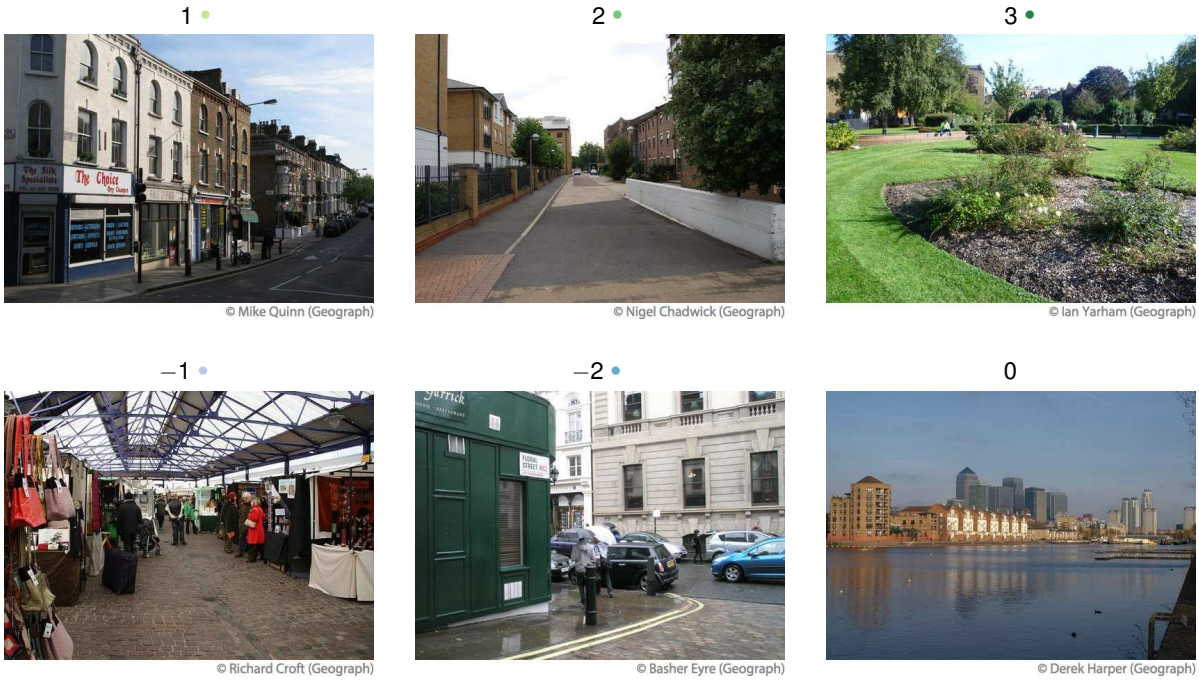


Figure 4.63: Examples of photographs with different values of the manually assigned greenness score. Positive numbers are used for 'natural' greenness and negative numbers denote man-made green objects. The images are courtesy of their authors and are shared under *Creative-commons BY-SA 2.0* licence.

	Flickr		Geograph		Panoramio		Σ	
3 •	8	3%	19	6%	13	4%	40	5%
2 •	22	8	65	22	52	18	139	16
1 •	20	7	46	15	39	13	105	12
0	211	72	163	55	184	62	558	62
–1 •	18	6	6	2	3	1	27	3
–2 •	13	4	1		5	2	19	2
–3 •	0		0		0		0	

Table 4.3: Assessed photographs grouped by the manually assigned greenness score.

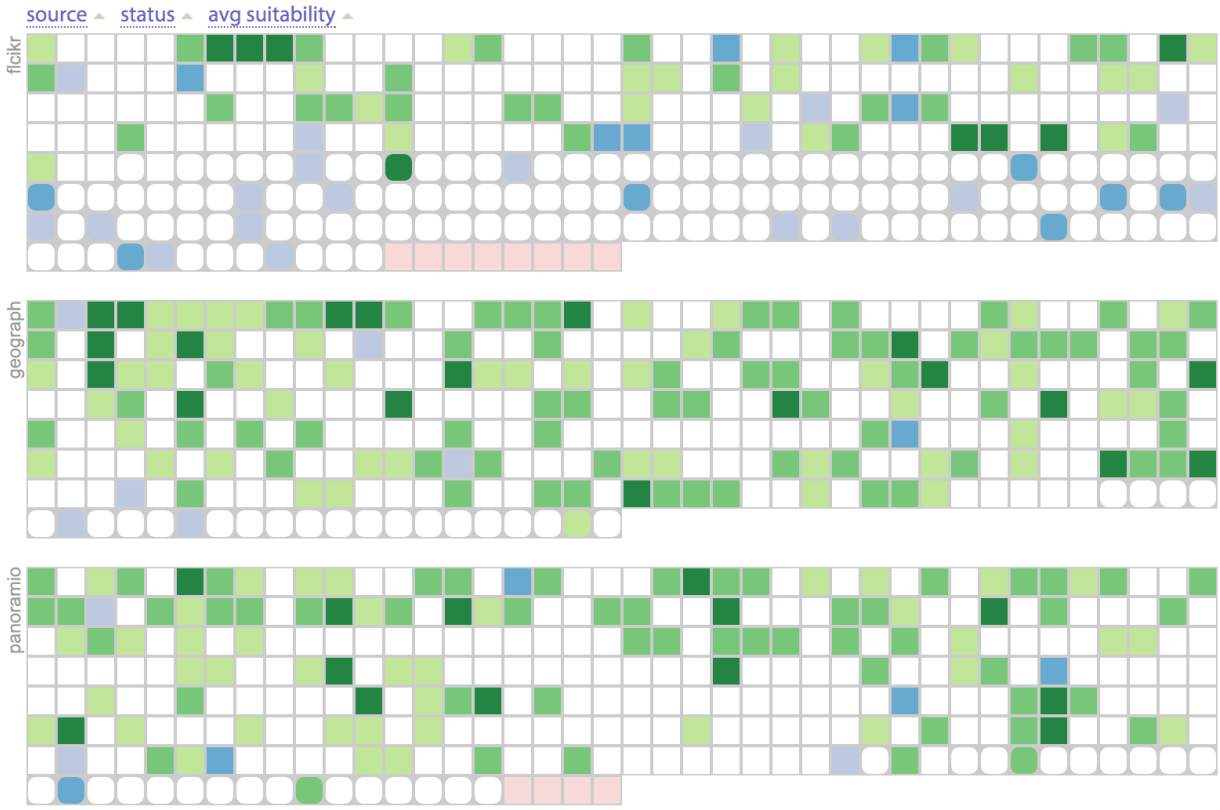


Figure 4.64: Row-prime ordered lists of assessed Flickr, Geograph and Panoramio images coloured by the manually assigned greenness score. Green cells denote photographs with presence of 'natural greenness' (trees, grass, etc.), and blue cells correspond to images where green objects have human-made origin. The darker the color the higher number of green pixels on the photo. Pale pink is used for images that were deleted at their origin after the launch of the survey, so had to be removed from the sample. Items with rounded corners are the photographs that were classified as taken indoors by the majority of respondents. Applied sorting mode is shown at the top of the figure.

4.6 Summary of all applied filtering methods

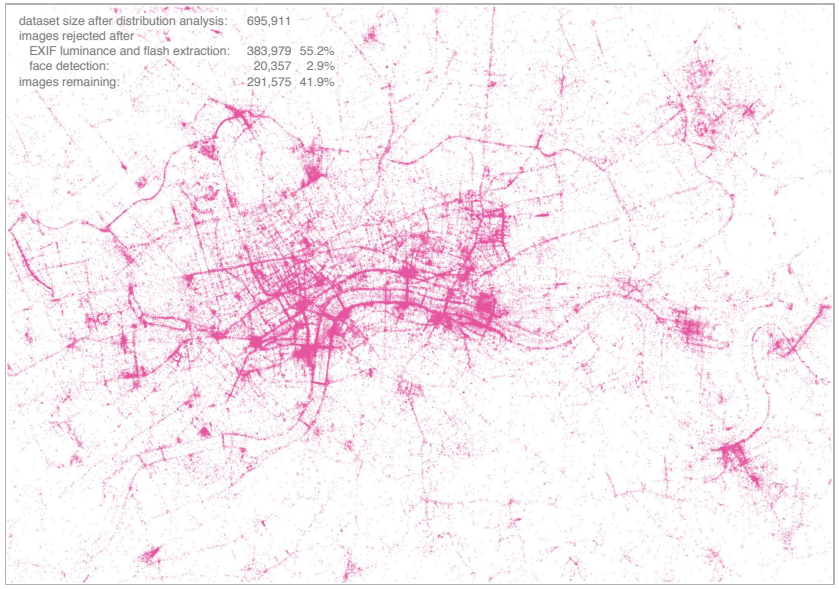
By analysing image metadata and content it was possible to find three new ways of reducing discrepancies between the existing crowd-sourced photographic collections and a *model photographic collection*, a concept of which is described on page 38. These were filtering by EXIF metadata (extracted luminance and flash usage), moderation category and presence of human faces. Cleaned versions of datasets are shown in Figure 4.65 on the next page. Because it was found that none of the chosen approaches could significantly improve Geograph data, only Flickr and Panoramio distributions were affected. The proposed methods became a useful addition to distribution-based data-cleaning techniques, described in Section 4.3.

The photo assessment survey, which helped collect human opinion on the quality of potential 'votes' for street attractiveness within random samples of photographs, was found a very useful instrument for suggesting metadata- and content-based filters. Combined with a visual analytic approach to data analysis, gathered subjective information allowed a number of hypotheses to be easily proposed and tested. Importantly, the survey also revealed the proportions of various kinds of images in the considered datasets and facilitated the evaluation of the results of cleaning. For example, subjective human classifications confirmed that exclusion of photographs with low values of luminance substantially increased the proportion of daytime outdoor photography in both Flickr and Panoramio collections. This and other approaches to filtering lead to less bias in street attractiveness scores computed from their spatial distributions.

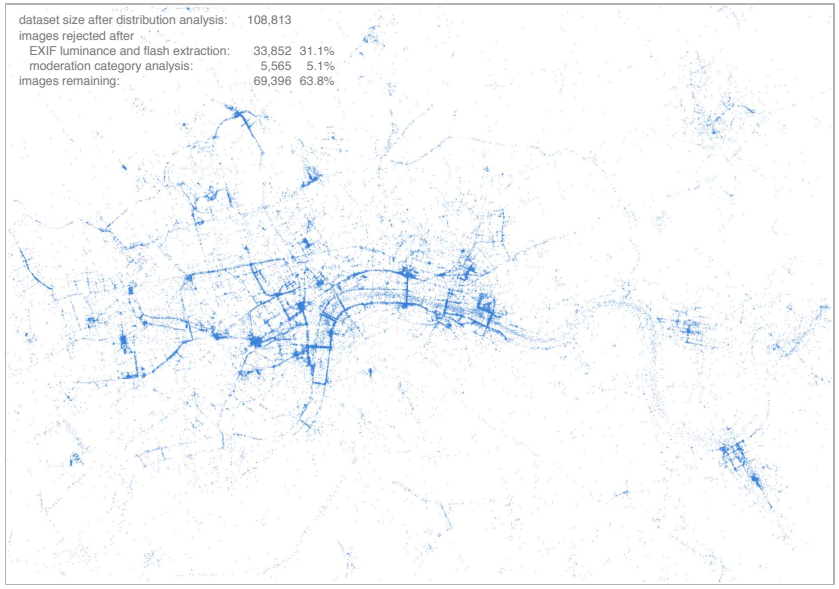
Table 4.4 on page 216 shows how the whole chain of applied binary filters affected the latest versions of the studied photographic collections. The table aggregates numbers attached to geographical maps in Figure 4.18 on page 136, Figure 4.42 on page 175 and Figure 4.65 on the next page. It demonstrates the scale of data processing that may be required to prepare original crowd-sourced photographic collections for their use in a route-suggesting system, in line with the theory described in Section 1.2 on page 17 and Subsection 3.1.2 on page 49.

This research does not claim that the proportions of removed photographs are always optimal, and that the bias in the distributions of 'votes' for street attractiveness is reduced in the best

(a) Flickr



(b) Panoramio



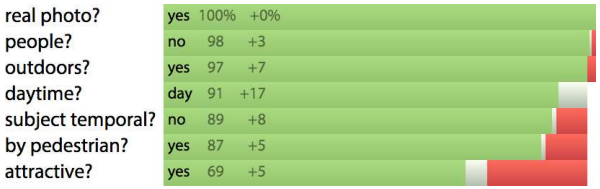


Figure 4.65: Images that remained in the latest versions of photographic datasets after metadata-based and content-based filtering. Distributions are supplemented with approximate proportions of different kinds of photographs according to filtered survey data.

	Flickr		Geograph		Panoramio		Picasa	
image records collected	2,219,354		77,464		183,735		1,121	
taken too recently	141,684	6%	4,648	6%	5,571	3%	1,081	96%
date of photographing is not known	0		0		30,575	17	0	
initial datasets	2,077,670	94	72,816	94	147,589	80	40	4
filtered by temporal coordinate	293,264	13	5,859	8	30,873	17		
filtered by spatial coordinates	1,072,782	48	7,874	10	7,630	4		
filtered as taken at local events	15,713	1	53		273			
remaining after distribution-based filtering	695,911	31	59,030	76	108,813	59		
filtered by EXIF data	383,979	17			33,852	18		
filtered by moderation category	0				5,565	3		
filtered by presence of human faces	20,357	1			0			
remaining after metadata-based and content-based filtering	291,575	13			69,396	38		

Table 4.4: Summary of all filtering methods applied to the latest versions of considered photographic collections (data were gathered on 2014-07-01).

possible way. Adjustment of some of the filtering methods has not been entirely formalised, which leaves an opportunity to explore more ways of their improvement. Nevertheless, the experiments have clearly demonstrated how different the content from various photographic sources may be and how important it is to transform the data in case if the intended utilisation of them does not match the original designation. Only 13% of all harvested Flickr records have passed all proposed filters compared to 76% and 38% of items from Geograph and Panoramio, respectively. Such significant disparity can be explained by a diversity of common patterns in the photographers' behaviour, which in turn are the result of the policy, the interface and the infrastructure of each photo-sharing website. The numbers may vary from city to city and are expected to be especially different in areas with a strong seasonal bias or with another proportion between casual and regular users.

The original image data filtering model, proposed in Subsection 3.1.1, split all potential binary filters into two types: (a) those that make a decision based on some knowledge about the whole dataset and (b) those that reject or accept a photograph solely based on its own properties. It was then suggested that all filters of *type b* can be applied after those of *type a* in order to make the whole process less resource intensive. Analysis of real data has revealed that a strict following of this principle may have a disadvantage in one case. Hotspots of the first type

(spatial coordinates with more than one photograph from a single user) can be better cleaned *after* metadata- and content-based filtering. For example, when a photographer attaches three images to a single spatial coordinate, and one of them is taken at night, another is a human portrait and the third is a daytime panorama, preserving one of them 'randomly' (e.g. by simply keeping the most recently taken photograph) may not be always appropriate. The solution to this issue can be in placing this particular part of spatiotemporal filtering to the very end of the chain of filters. This operation, however, requires expensive data analysis to be executed for a much larger photographic collection, and there is no guarantee that improvement in the distribution of 'votes' will be significant. A detailed study of this issue has not been included into this research project.

The concepts of bias-reduction function $B(C)$ and mapping function $M(e,B(C))$, which were introduced on page 21 as steps of getting street attractiveness scores A_e, have been also slightly revised during the analysis of photographic data. Examination of individual contributions from active photographers in all studied datasets (Figure 4.21 on page 141) suggested that it is more reasonable to count 'voted' users, not the image records themselves when converting spatial distributions of images into the network edge scores. This can guarantee that requirement 5 (page 38) is always met with no need to apply complex methods of removing items that are located within a few meters from each other and are owned by the same user.

Existence of several alternative filtered collections of 'votes' for street attractiveness raises a reasonable question about which of them may be more suitable for informing a leisure routing algorithm. The nature of the problem suggests that this question can be confidently answered only after the paths have been finally generated by the system and then subjectively evaluated by a crowd of pedestrians 'in the field'. However, some judgement can be still made beforehand by looking at Pareto efficiencies of the available choices (Chinchuluun et al. 2008). The requirements for a *model photographic collection* (defined on page 38) can be used as criteria for the comparison.

In the given group of Flickr, Geograph and Panoramio datasets, two were found Pareto-efficient. According to the photo assessment survey data, Panoramio contained the highest proportions of valid 'votes', both before and after metadata- and content-based filtering. Flickr was Pareto-efficient, because the final number of photographic records it contained was the

largest, even despite that 87% of them were removed during the bias-reduction process and the quality of the remaining items was lower than of those from Panoramio. More 'votes' for street attractiveness from a bigger crowd of photographers gave a stronger support to requirements 4 and 6 (page 38). This feature of the data implied less personal bias and potentially fewer road segments with scores equal to zero. Filtered Geograph dataset belonged to a Pareto frontier only when compared to Flickr – it had fewer data records, but a higher proportion of valid 'votes' for street attractiveness (see diagrams in Figure 4.52 on page 185 and Figure 4.65 on page 215 for comparison). However, it was further from a *model photographic collection* than Panoramio by all 10 requirements. This suggested not to involve Geograph in further experiments within the limits of this research project.

Chapter 5

Experiments with road network data and routing

In line with research questions and research workflow, this chapter focuses on linking the problem of pathfinding with the distributions of 'votes' for street attractiveness from cleaned crowd-sourced photographic datasets. Described experiments correspond to two components of a photo-based routing system for leisure walks, a general structure of which can be found in Figure 3.1 on page 36.

The first set of experiments is dedicated to the process of score-to-edge assignment, involving a description of a general approach to this task and a number of comparison tests. Second, the proposed routing algorithm is discussed. This part of the chapter contains information about its implementation and also lists a number of derived findings. Finally, the chapter questions the problem of route evaluation and other relevant issues.

Before any of the above experiments could be conducted, it was necessary to obtain the arrangement of walkable streets within the chosen region (the boundaries are shown in Figure 3.6 on page 59). Subsections 5.1 and 5.2 contain a brief summary of steps that were taken to achieve this goal.

5.1 Selection of a source for the road network data

Almost any available source of road network data could be suitable for the experiments in this research. A graph with locations of urban walkways did not have to be either 100% accurate or reflect all the latest changes in the configuration, such as temporary closed streets (this would only be important if it was intended to make the routing system available to the public). It would be preferable, however, if the data were pedestrian-oriented, i.e. excluded road segments where walking is forbidden and included dedicated footways, such as ones in the parks.

The question of which source of data to use was not answered uniformly in related research. For example, Zheng et al. (2013) and Quercia, Schifanella and Aiello (2014) presumably used data from Google, and Alivand and Hochmair (2013) worked with a street network provided by TomTom. After considering these and several other options it was decided to choose OpenStreetMap (`http://osm.org/`) as a base for a road network graph in this project. According to OSM-Wiki (2014*b*), in a number of recent years this free and open cartographic database was integrated into a great variety of routing services all over the world, which could be considered as a good a sign of its suitability for the chosen purpose. Road network quality in OpenStreetMap was analysed in a number of studies (e.g. by Haklay 2010; Hochmair and Zielstra 2011; Antoniou 2011), where this source of data was compared to governmental and proprietary databases. The conclusions of these works confirm that OpenStreetMap could be considered as appropriate for network-related experiments in this research. Furthermore, manual exploration of OpenStreetMap in London revealed an even more detailed coverage of walkable paths than in its alternatives:

(a) Google Maps (b) Navteq (c) Ordnance Survey OpenData (d) OpenStreetMap

Figure 5.1: Comparison of footpath coverage in different map data (2013, near Regents Park).

OpenStreetMap is maintained by a self-organising community of volunteers rather than a company that takes responsibility for the data (Ramm, Topf and Chilton 2011); this was both a potential flaw and an advantage for its use in the research. On one hand, such fact increased the chances of local errors, e.g. disconnected road segments or inaccurately assigned access restrictions for pedestrians. As a consequence, the results of routing could become unexpected and harder to investigate in some situations. On the other had, openness of the data made it possible to fix any errors at the origin and then download a new version of the road network for another round of experiments. With a range of third-party quality assurance tools available for OpenStreetMap such as KeepRight (`http://keepright.ipax.at/`), OSM Inspector (`http://tools.geofabrik.de/osmi`), Osmose (`http://osmose.openstreetmap.fr/en/map`), etc. most of the issues with routing could be located and corrected. This was done for Central London in 2013 before starting the second part of this research project.

OSM data are licenced under ODbL (OpenStreetMap 2012).

5.2 Road network data gathering

Being a general-purpose collaborative project, OpenStreetMap accumulates information about various geographical objects including administrative boundaries, land use, buildings, vegetation, street furniture, etc. Standard OSM API (`http://wiki.osm.org/wiki/API_v0.6`) allows developers to retrieve map data for an arbitrary defined bounding box, but is not giving an opportunity to narrow the queries down by the object category. This makes it difficult to fetch road networks for areas greater than several square kilometers, because the total amount of data to transfer exceeds the limits of the API. It becomes necessary to (1) divide the query geographically, i.e. make several queries for smaller regions, (2) merge the obtained data and (3) clean the dataset from all irrelevant objects.

A number of third-party services provide a solution for the above complexity. Their similar read-only *extended* APIs (`http://wiki.osm.org/wiki/XAPI`) give access to copies of the OpenStreetMap database and make it possible to define filters as parameters of spatial queries. Thus, only certain kinds of objects in the area if interest may be retrieved. There are still

limits on the amount of data that can be harvested with a single request, however these limits are significantly higher than it might be necessary for obtaining all walkable paths in a city. A disadvantage of using an extended API instead of the original one is that the data may not be always up-to-date. After something has been changed in the OpenStreetMap database, it needs to be pushed to the third-party server and indexed, which creates a delay of up to a few days.

According to the naming convention, all roads in OpenStreetMap (including footways) have tag `highway` (OSM-Wiki 2014*a*). Therefore, retrieving the entire network of roads in Central London (Figure 3.6 on page 59) can be done with a single query:
`http://api.openstreetmap.fr/xapi?*[highway=*][bbox=-0.21,51.46,0.02,51.56]`
(`api.openstreetmap.fr` is one of existing XAPI servers).

In order for a downloaded network of roads to become navigable (i.e. in a format that most of the routing algorithms can work with), it has to be converted into what is called *topology*, or a *routing graph*. Polygonal lines, which represent streets, are divided into segments at each intersection, and all junctions are assigned with unique identifiers. As such task is rather common, there exist a number of free tools that help perform the transformation. In this research, it was chosen to use OSM2PO (`http://osm2po.de/`) – this tool converts OpenStreetMap data in its original `*.osm` format into a PostgreSQL / PostGIS table.

Depending on the mode of transportation that a routing system aims to support, some roads have to be excluded from the original network when it is being converted to a topology. For example, footways, cycleways and pedestrian-only streets need to be removed for car navigation (alternatively, such road edges can be assigned with an infinite cost). In this research, OSM2PO was configured to keep all types of roads except those that were marked by OpenStreetMap volunteers as unsuitable for pedestrians, i.e. motorways, dedicated cycleways, private streets, etc. (`wtr.finalMask = foot` in `osm2po.config`).

All urban footpaths are usually interlinked, so that pedestrians can walk from one arbitrary place to another with no need to switch between the transportation modes. However, due to errors in data, some street segments in a harvested topology may be isolated, thus forming 'islands' of roads. As these 'islands' are not connected to the majority of the edges, it becomes impossible to generate a route between *any* two randomly chosen nodes, which consequently

causes software errors. This issue can be avoided by dividing the topology into a set of interconnected 'islands' and then excluding all edges that do not belong to the largest one.

Another group of edges that can be removed from the topology are those that do not geographically intersect with the area of interest. If the original dataset has 'hair' boundary (i.e. the objects are not sliced at the verge of the bounding box), some topology edges may end up being completely outside of the chosen region. This situation happens every time when harvested roads intersect *not* within the bounding box. Unlike 'islands', external edges may not cause critical errors in the routing algorithm.

Figure 5.2 shows the latest version of the topology for Central London that was collected from OpenStreetMap and then used in the experiments.

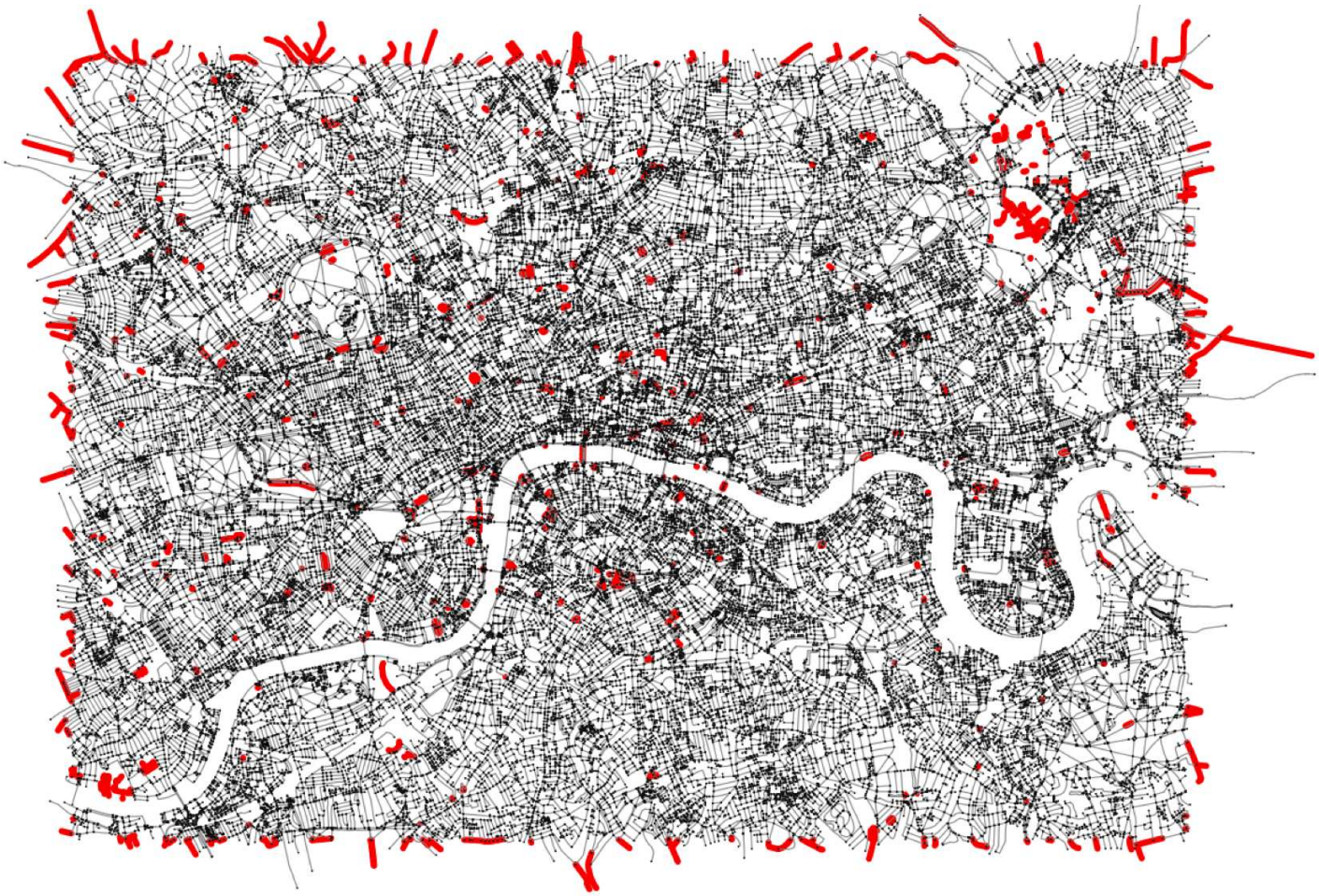


Figure 5.2: Road network topology for Central London (December 2013). Red paths correspond to isolated edges ('islands') and those edges that are outside of the chosen region. Both were excluded from the routing graph. *Road network © ODbL OpenStreetMap contributors.*

With the Dataset Abstraction Framework (Section 3.2), topology gathering and cleaning could be performed the following way:

```
$ cd photo-routing

# Initialise domain 'networks' that will contain one or more datasets with road
    networks
$ app/console da:domain:init networks

# Initialise dataset 'london_osm_dec2013', download the network from OpenStreetMap,
    create the topology and save it as "topology_raw" component of the new dataset
$ app/console pr:networks:create-from-osm london_osm_dec2013
    "bbox(-0.21,51.46,0.02,51.56)" --no-cache

# Find topology edges that are outside the dataset's bounding box
$ app/console pr:networks:topology_raw:find-edges-outside-bounds london_osm_dec2013

# Find topology edges that form 'islands' (i.e are isolated from the rest of the
    network)
$ app/console pr:networks:topology_raw:find-islands london_osm_dec2013

# Move all edges from component 'topology_raw' to 'topology_edges' (except for
    isolated 'islands' and edges outside the bounding box); populate 'topology_nodes'
$ app/console pr:networks:topology:extract-from-raw london_osm_dec2013
```

A general structure of a road network dataset is show in Figure 5.3. Because in this project a topology could be only sourced from OpenStreetMap, there was no need to introduce dataset types with type-specific attributes or components – all datasets belonging to domain `networks` were homogeneous.

Derived component `topology_edges` contained all information that was required for experimenting with route generation. Attractiveness scores A_e (see Formula 3.2 on page 3.2) were stored in attributes `score__[window-type]_[window-size]__[photoset-name]__[subset-id]`. This made it possible to calculate street attractiveness based on arbitrary subsets of records from different photographic sources and also test various window designs with no need to copy the topology. Real costs of edges (their length l_e) were stored in attribute `length`.

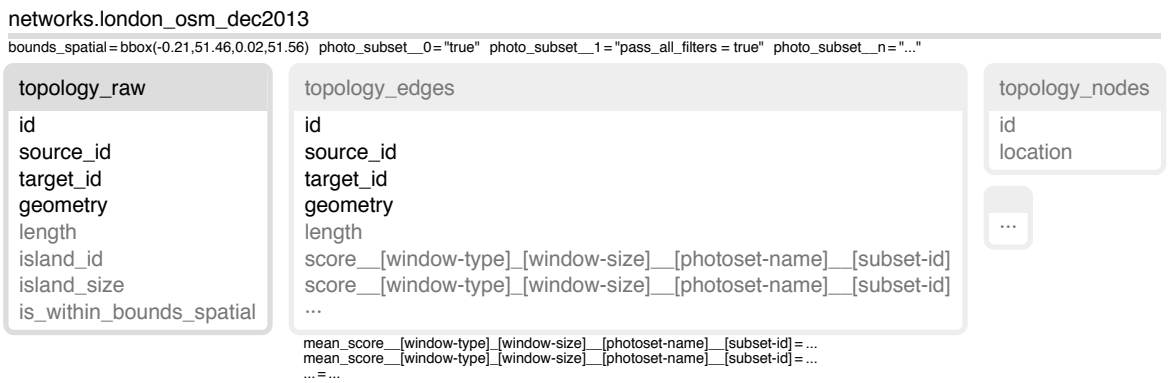


Figure 5.3: General structure of a road network dataset.

All experiments that are mentioned in this chapter have been conducted with use of a single version of the road network. After cleaning (i.e. removing 1,289 edges that were isolated or were outside the bounding box), the topology contained 87,743 edges and 65,394 nodes. The longest edge was 2,193 meters, and the shortest one was only 32 centimeters. Average edge length was 48.7 meters, and the median was equal to 34.6 meters. A distribution of topology edges by their lengths is shown in Figure 5.4.

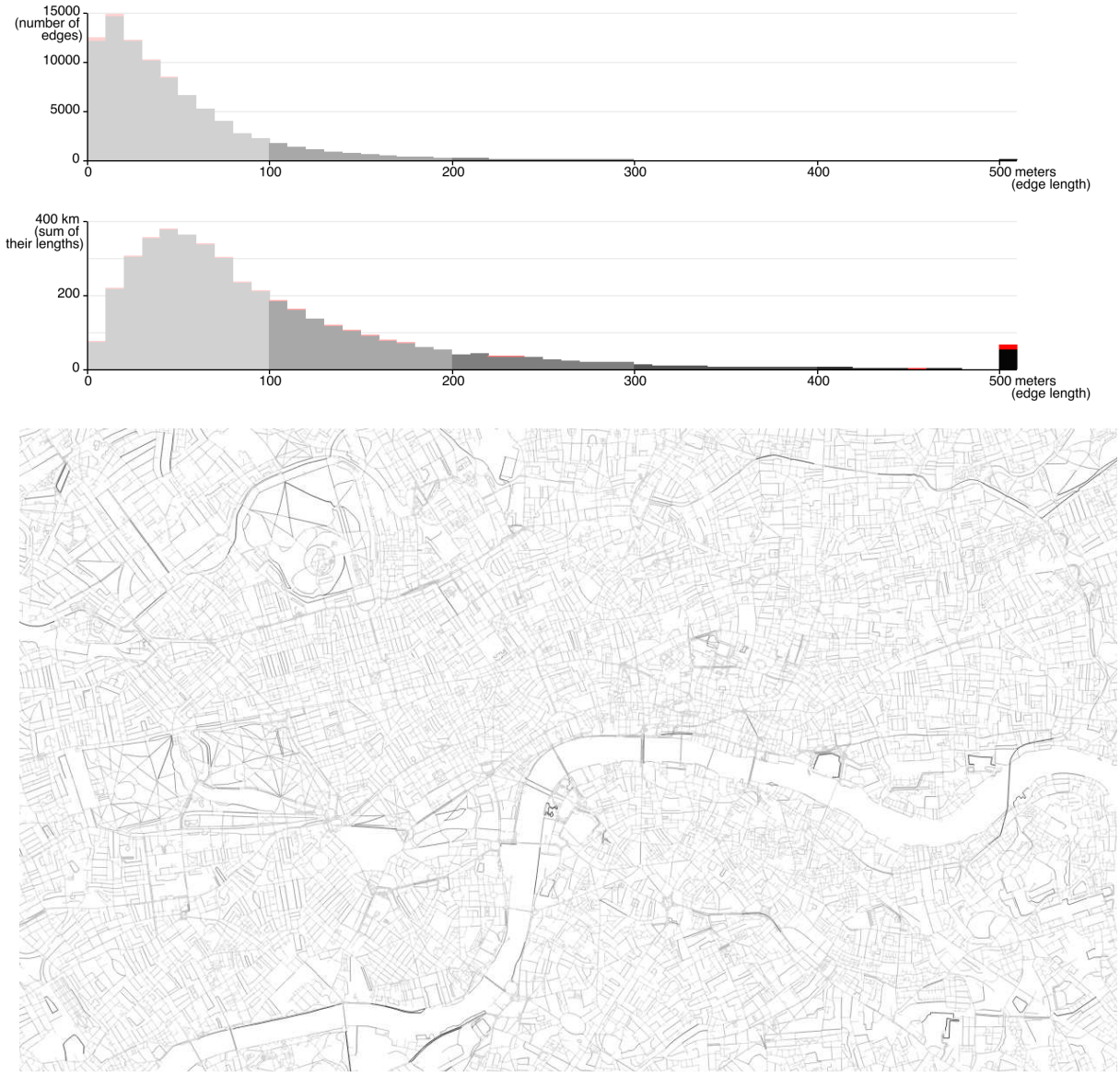


Figure 5.4: Topology edges by their length. For compactness, the map shows only a part of the considered region. Red sections of bars represent edges that were previously excluded from the graph. *Road network © ODbL OpenStreetMap contributors.*

5.3 Street attractiveness scores

5.3.1 Assignment of attractiveness scores to road network edges

According to the introduced theory (page 21) and methodology (Subsection 3.1.2 on page 49), edge attractiveness scores are computed based on the locations of neighbouring geotagged images. The result depends on the *shape of the edge window* within which the photographs are taken into account, the *original photographic dataset* and the *bias-reduction function* that has been applied to remove irrelevant 'votes'. Thus, to extract the scores, it becomes necessary to obtain mutual positions of the photographs and road segments.

The process of calculating the distance between a photograph (a point) and an edge (a polygonal line) is non-trivial, especially if the geographical coordinates of these objects belong to a spheroid, not a projected plane. A number of trigonometric functions need to be involved in both cases, which makes the operation rather resource-intensive. Given that an urban area may be covered by millions of geotagged images and contain hundreds of thousands road edges, finding neighbouring entities may take unacceptable amount of time. Besides, with any repetition of the process of score assignment (e.g. for a different window size), distance extraction must be redone if the task is being approached straightforwardly.

An efficient algorithm for score-to-edge assignment was proposed and tested during this research. Its idea is in dividing a geographical region into quads, introducing a concept of a 'photo window' and splitting the process into two independent stages.

The goal of the first stage is to calculate the distances from all potential 'votes' in a given photographic dataset to the neighbouring topology edges and nodes (i.e. the objects that are not further than the maximum size of an edge window that is going to be used in subsequent experiments). First, the bounding box that contains the entire considered road network is divided into quads of equal size (see Subsection 3.2.4 on page 83 and Subsection 3.3.3 on page 99). Then, the following actions are performed for each quad:

1. Find all image records that belong to the concerned photographic dataset and are located within the quad; load their identifiers and locations into the memory.

2. Expand the quad by the *maximum edge window size* (e.g. 200 meters).

3. Find all topology edges that are fully or partially located within the extended quad; load their identifiers and shapes into a temporary table.

4. Find all topology nodes that are located within the extended quad; load their identifiers and locations into another temporary table.

5. For each image record in the memory, calculate Cartesian distances to all edges in the first temporary table; do the same for the nodes in the second temporary table. Save identifiers of edges and nodes that are closer to the image record than the *maximum edge window size*.

6. Delete both temporary tables.

Consequently, a dataset that contains a road network is supplemented with one or more derived components as shown in Figure 5.5. The number of these components correspond to how many photographic datasets it is planned to use for the assignment of street attractiveness scores in future experiments. The lists of edges and nodes that are spatially located near an image have been called *photo windows* by analogy with *edge windows* – areas around the polygonal lines (walkways) that contain 'votes' (see Figure 3.5 on page 50).

With DAF, photo window extraction could be done the following way:

```
$ app/console pr:networks:photo-windows:calculate london_osm_dec2013
    london_panoramio_jan2014 --reset-all --gui --thread-count 20
# london_osm_dec2013: the name of a dataset that contains the road network
# london_panoramio_jan2014: the name of a photographic dataset to work with
# --reset-all: cleans previously extracted photo windows if necessary
# --gui: shows a dialog with a visualization of the progress (see subsection 3.3.3)
# --thread-count: 20 sets the number of parallel threads to 20 (default is 10)
```

In Central London, it took approximately a quarter of an hour to extract photo windows for the initial Panoramio data and about an hour for all gathered Flickr image records when the size of a quad to process was set to one square kilometer, and the process was distributed among ten parallel threads.

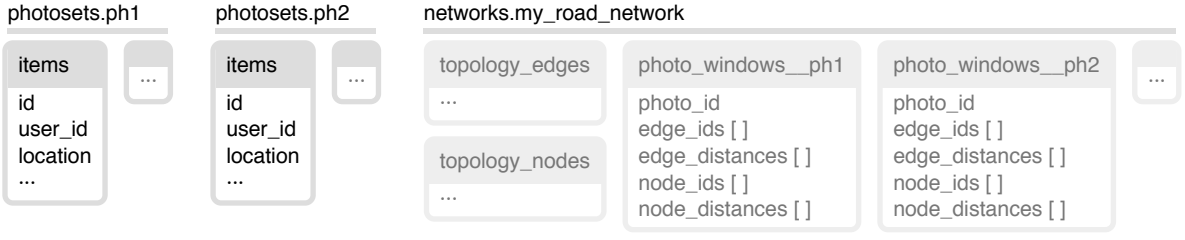


Figure 5.5: Photo windows in a road network dataset.

After the distances between all photographs and their neighbouring edges and nodes are pre-calculated, actual score extraction becomes relatively inexpensive. There is no more need to involve resource-intensive trigonometric functions, because all potentially required intervals can be extracted from a derived component. Thus, it becomes easy to experiment with various window shapes, vote counting methods and bias-reduction functions.

The second stage of score-to-edge assignment also benefits from dividing the area into smaller parts and parallel data processing. A sequence of actions for each quad is the following:

1. Find all topology edges that are located within the quad, load their identifiers and the identifiers of their end nodes into the memory.
2. Expand the quad by the currently chosen window size (see details below).
3. Find all photographs within the obtained extended region that match the chosen data filter (i.e. bias-reduction function); supplement their own identifiers and the identifiers of the photographers with the data from a corresponding *photo windows* component.
4. For each edge, loop though all loaded photo windows, find those that belong to the *edge window* and then convert a subset of selected ‘votes’ into a score.

Quad expansion at the second stage of score-to-edge assignment is more complex than during the photo window forming, where the bounding box is simply padded with a fixed number of meters (*maximum edge window size*). In order for all the scores within a quad to be counted correctly, it is necessary to load all photo windows that are nearby all selected edges including those that are partially outside of the bounding box. Thus, the shape of the expanded quad should contain prominences, as shown in Figure 5.6 on the facing page. The process of quad expansion consists of these four steps:

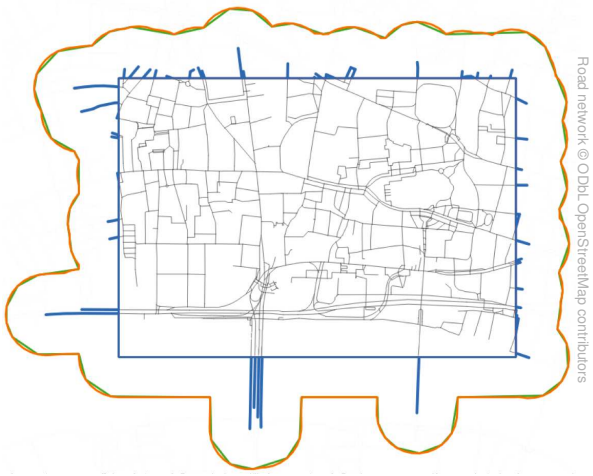
1. Find edges that are partially outside of the quad; make a small spatial buffer around them.
2. Combine this buffer with the original bounding box.
3. Make a buffer around the obtained area; its size should be equal to the current edge window size.

4. Simplify the result using Douglas-Peucker algorithm (Douglas and Peucker 1973) with tolerance of not more than 1 meter. The reduction of the complexity of a polygon increases the performance of a spatial query that fetches photo windows.

The smaller the quads, the fewer edges and photo windows are loaded into the memory at a time, however the more is the proportion of edges, for which the scores are calculated more than once due to their belonging to two or several quads. Thus, working with the areas that are too small does not help improve the performance of the process.

A number of tests have suggested that when the quads are about one square kilometer in size, the performance of the second stage of score-to-edge assignment is one of the highest. With this parameter and ten parallel threads, it took about two minutes to calculate a single score for all network edges in Central London based on Panoramio data; for Flickr the process took approximately five minutes. With DAF, new scores could be calculated by calling a single command:

```
$ app/console pr:networks:scores:calculate london_osm_dec2013
    uc_50__london_flickr_jul2014__initial,
    ucn_b_e_f_100__london_panoramio_jul2014__pass_all_filters --reset-all --gui
    --thread-count 20
# london_osm_dec2013: the name of a dataset that contains the road network
# uc_50__london_flickr_jul2014__initial: user count, window size = 50 meters,
    photoset: latest flickr,  bias-reduction function: none (all photographs in the
    initial dataset are considered)
# ucn_b_e_f_100__london_panoramio_jul2014__pass_all_filters: user count normalised
    (divided by edge length) + '_b' for window blurring + '_e' for exclusion of ends +
    '_f' for vote fission, window size = 100 meters, photoset: latest panoramio,
    bias-reduction function: complete (all successful filtering methods from chapter
    four are applied)
# --reset-all: cleans previously extracted values if necessary
# --gui: shows a dialog with a visualization of the progress (see Subsection 3.3.3)
# --thread-count: 20 sets the number of parallel threads to 20 (default is 10)
```



grey: network edges in the considered quad

blue: buffer around the bounding box and the edges with segments outside it

orange: extended region

green: simplified extended region

Figure 5.6: Expansion of a quad when calculating edge scores.

All spatial functions that were involved in both stages of score-to-edge assignment (such as distance calculation, buffer creation, union of polygons, etc.) were delegated to PostGIS. This helped reduce the amount of work and ensured the correctness of the results. The only geography-related task that could not be done naively by PostGIS was exclusion of ends from the edge window (see Figure 3.5e on page 50). This could be achieved by comparing Cartesian distances from a photo window to a given edge and to both of its nodes. If a photo window is closer to the edge than to any of the two end nodes, it can be considered as belonging to a window with excluded ends.

The proposed two-stage approach was found fairly efficient. First, use of quads significantly reduced the number of required calculations and also lessened the memory footprint of the software tool. This made the solution applicable for a spatial region of any size, not only limited to a central part of a single city. Second, introduced pre-calculated *photo windows* made it easy to generate numbers of customised scores for the same combination of photographic data and road network data, thus enabling various comparison tests. Finally, the use of *photo windows* made it possible to implement all *edge window* designs listed in Figure 3.5 on page 50.

Two following subsections demonstrate the results of some score comparison tests for edge windows. Figure 5.7 explains the design of a visual data layout that is used in this report to support the discussion of the findings.

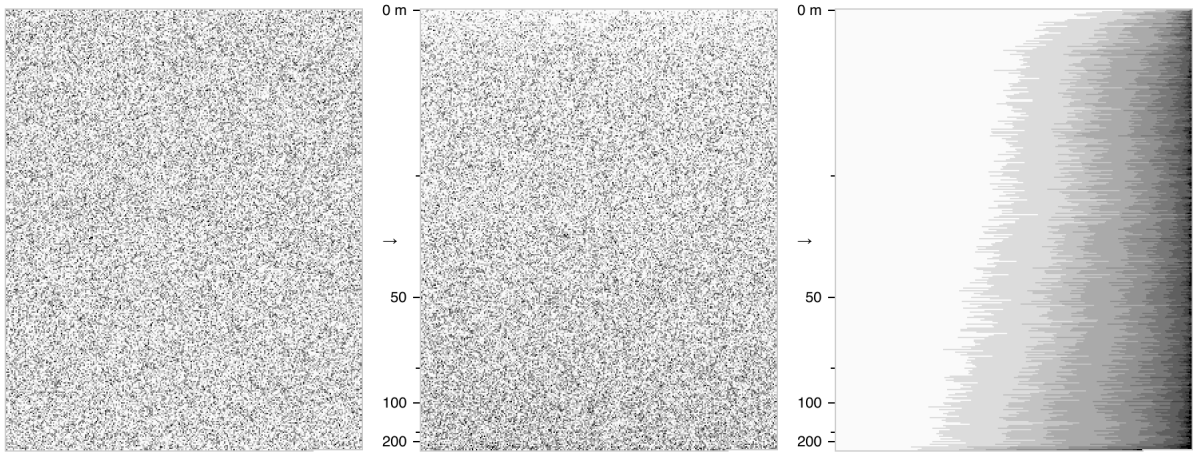


Figure 5.7: Visual representation of attractiveness scores for all 87,743 topology edges in Central London. *Left:* an unsorted set of all edges where each edge is represented with a single pixel and is coloured by the value of a score. *Middle:* The same set with row-prime ordering of edges by their length. *Right:* edges in each row are ordered by the value of the score.

Initially, attractiveness scores were derived for both Flickr and Panoramio datasets with use of standard edge windows (see Figure 3.5a on page 50). Window size (radius) was made equal to 50 meters, as suggested by Alivand and Hochmair (2013). The result is shown in Figure 5.8.

When edge window design is standard, attractiveness mapping function M (see Equation 1.2 on page 21) simply counts the number of unique users that have left any amount of 'valid votes' near a given edge. In this context, a 'valid vote' is a geotagged photograph that has passed all binary filters within the bias-reduction function B. If *overall photo count* was used instead of *user count* to form attractiveness scores, some values would be significantly biased by the most active individuals in both communities. More on this issue can be found in Subsection 4.3.1 on page 137.

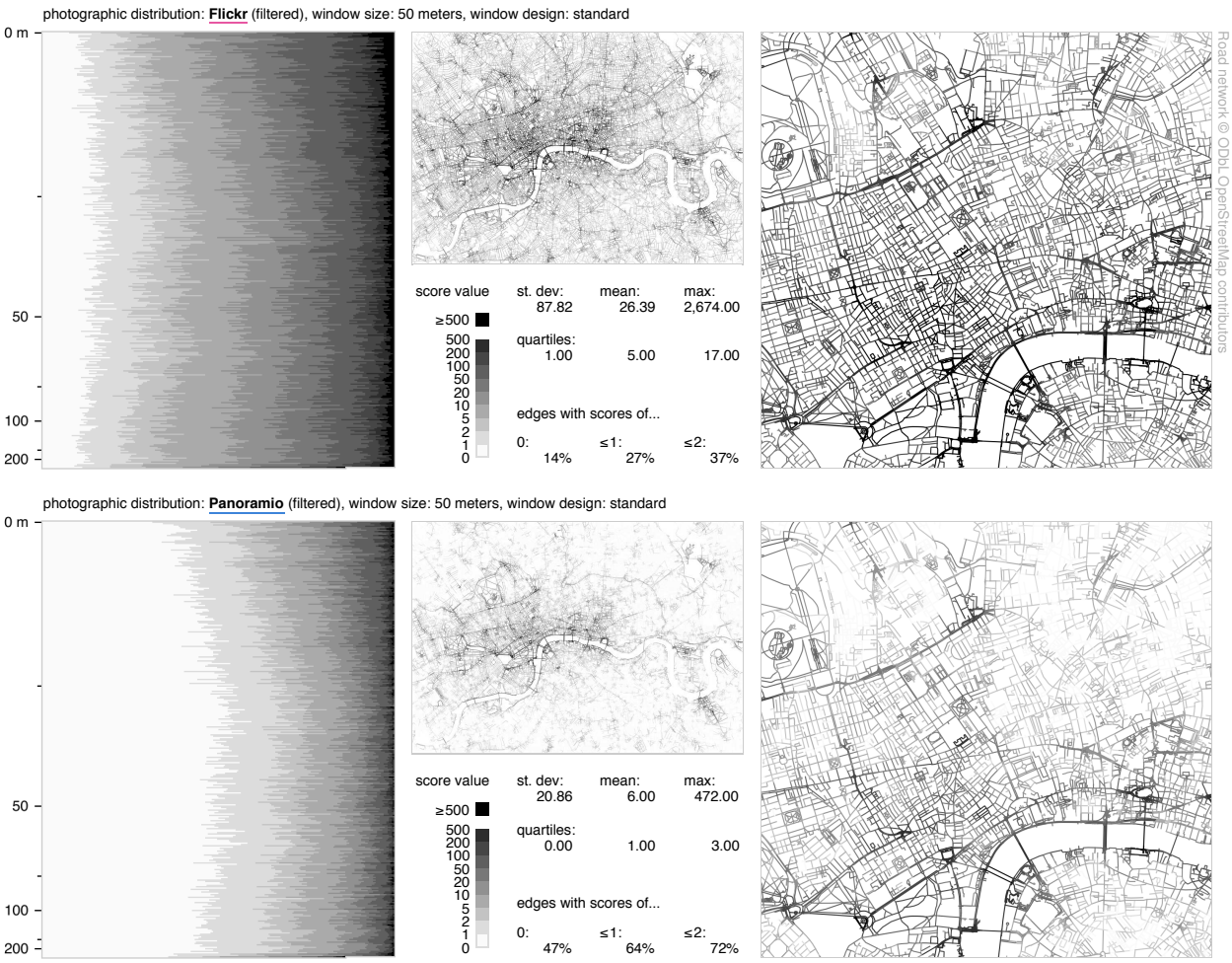


Figure 5.8: Initial street attractiveness scores.

5.3.2 Sensitivity to edge window design

With use of combined visual layouts and derived statistics, it was possible to assess the potential trustworthiness of the scores and also make comparisons of various edge window designs. This could facilitate a search for an efficient approach to 'vote' counting.

A quick look at initial edge attractiveness scores (Figure 5.8 on the preceding page) confirmed the existence of a correlation between their values and the spatial densities of photographs that passed all proposed filtering methods in Chapter 4 (see Figure 4.65 on page 215). According to the maps, the locations of the most 'valued' street segments were often concentrated around popular landmarks, near the water and in the parks – this was a good sign of the reliability of a photo-based routing system and supported the theory in Section 1.2. The diagrams, however, revealed a suspicious pattern: the edges of shorter length both in Flickr and Panoramio data were not having lower scores than longer edges, as it could be expected. Score normalisation by edge length (Figure 5.9) was able to emphasise the anomaly.

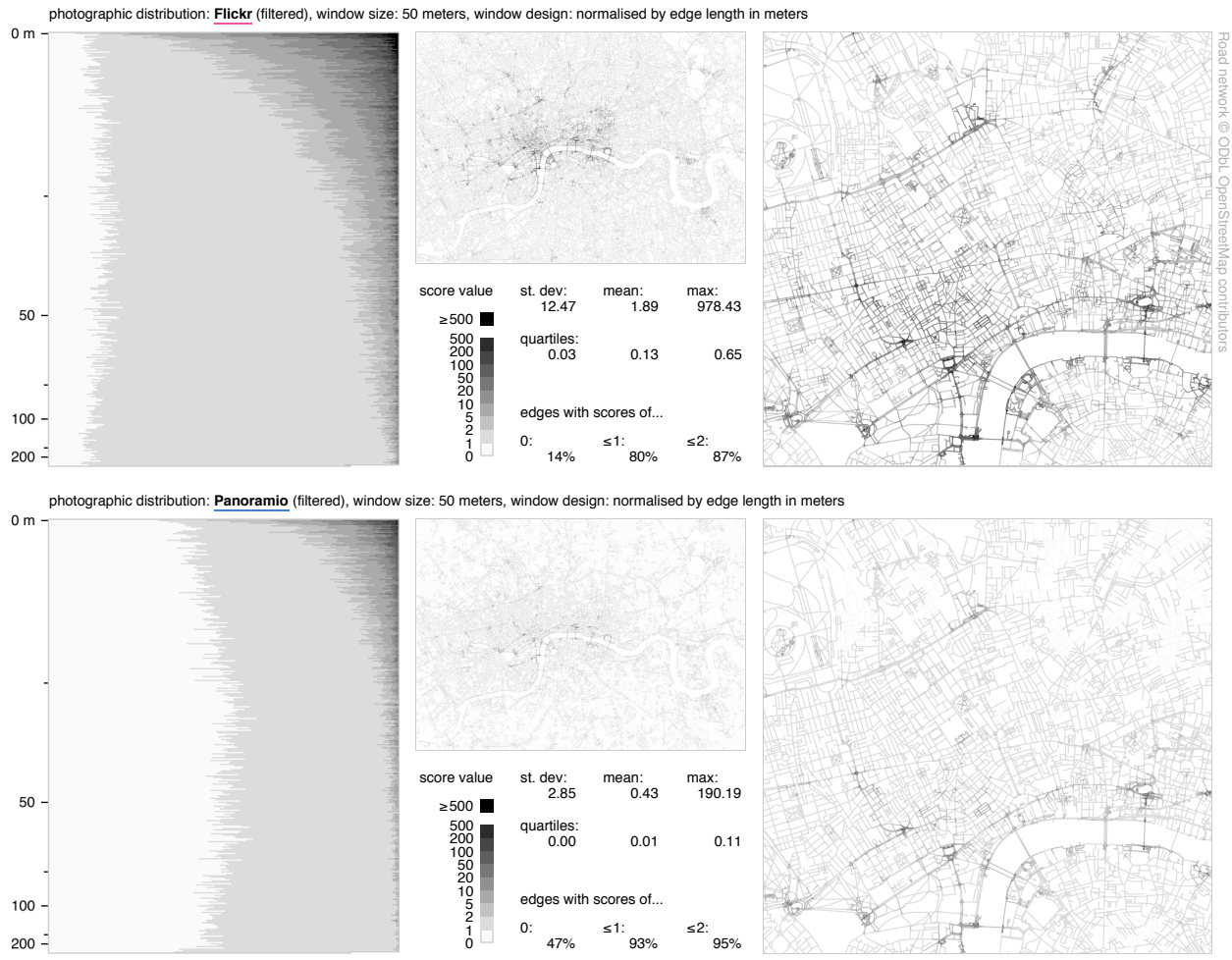


Figure 5.9: Initial street attractiveness scores normalised by edge length in meters and rounded to the smallest following integer.

This anomaly could be explained by two factors. The first one was a bias in locations of edges having different lengths. It appeared that the given street network topology had uneven level of detail, i.e. there were more short road segments near the city centre in general and around some popular attractions in particular. Places like Trafalgar Square and Picadilly Circus contained tens and hundreds of short linked walkways (crossings, pavements, stairs), which naturally increased the likelihood of small road segments to obtain higher scores. A proportion of long edges such as bridges, embankments or footways in parks were also located in areas that were popular among the photographers, but the vast majority of such roads were concentrated in residential areas (see Figure 5.4 on page 225). The second and the most influential factor, however, was a problem in the window design.

When a straight road segment has length l and a standard edge window has size s, the area of the window equals to $2ls + 2\frac{\pi s^2}{2}$ (see Figure 3.5a on page 50). The second summand is constant for any l, which means that a proportion between the edge length and the window area is not always the same. As a result, smaller road segments accumulate unreasonably higher numbers of 'votes' and thus distort their distribution. This effect can negatively influence a photo-based routing algorithm by making it sensitive to road segmentation. For example, if in a network there exist two equivalent footways surrounded by identical distributions of images, splitting one of these footways into two can make it more preferable. Indeed, the overall area of the edge windows will become higher by πs^2: $2ls + \pi s^2 \to 2(2\frac{l}{2}s + \pi s^2)$. Consequently, the 'votes' that are located near the split will contribute to scores of both halves of the divided footway and thus unreasonably increase the sum of gain.

Figure 5.10 on the following page shows what happens to the attractiveness scores when the ends are excluded from the edge windows, i.e. when all windows become smaller by πs^2 (the design is explained in Figure 3.5e on page 50). As it can be seen from the diagrams, the scores become more correlated to the edge length. The dependency however, is still not linear, and normalised scores continue to be higher for shorter edges. Apart from the natural source of this bias that has been mentioned earlier, two more factors have been revealed during the detailed analysis of the data. First, it has been discovered that end exclusion is not a 'silver bullet' for making the proportion between the length of an edge and the area of a corresponding window constant. Although the method works well for edges that are straight or nearly straight, it still

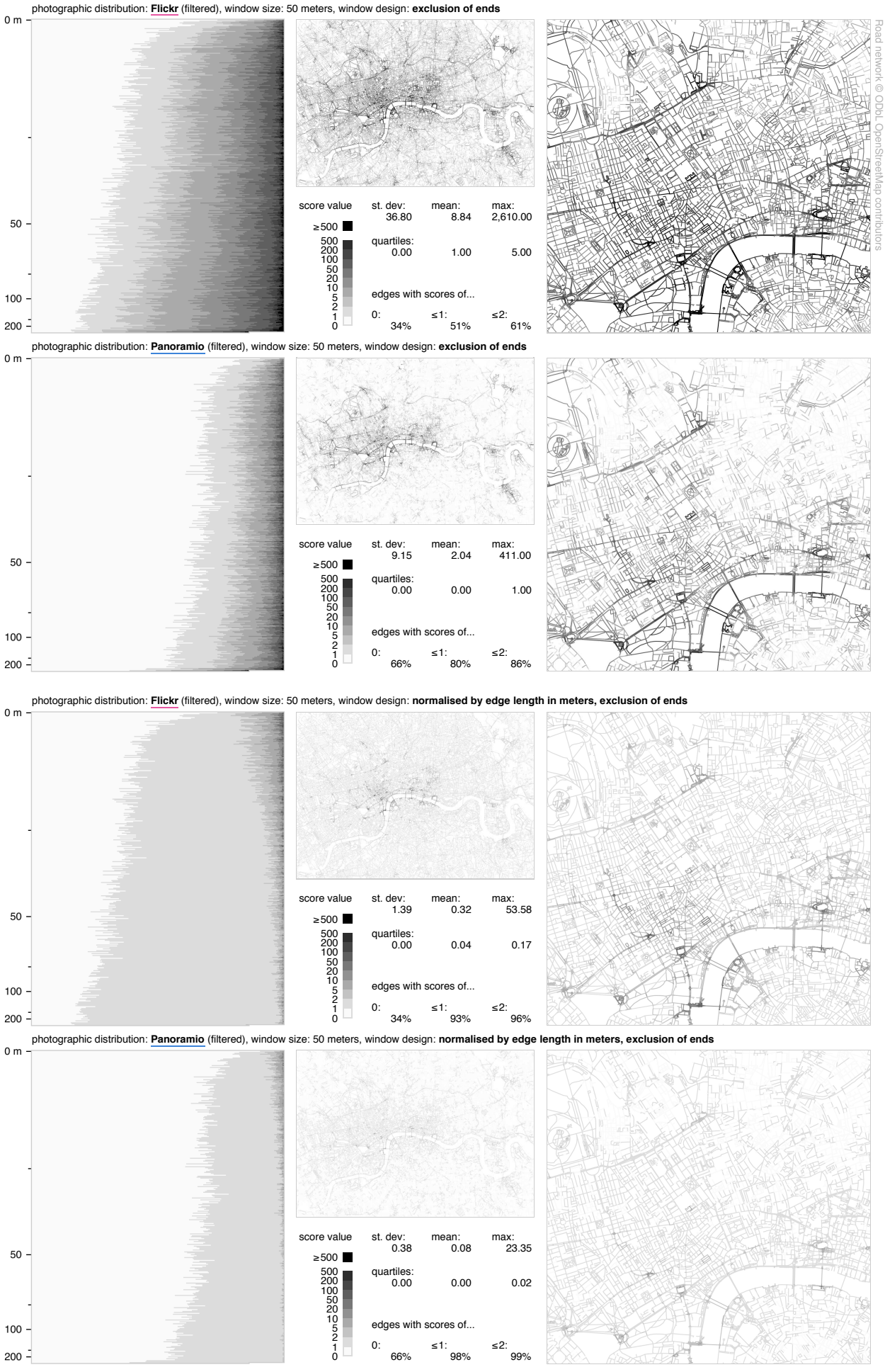


Figure 5.10: Effect of exclusion of ends in standard edge windows.

produces relatively large windows for short bended road segments (e.g. ⌒). This fundamental problem could be possible approached by making window size *s* dynamic or by splitting the topology edges at the sharpest bends. Another issue is related to dealing with long edges. Because the edge score is formed by counting the number of contributed photographers, not individual geotagged images, bridges, footways in the parks and other long roads may get smaller scores than they 'merit'. Indeed, if a single photographer leaves 'votes for attractiveness' near both ends of a 370-meter-long Millennium Bridge (which is represented with only one topology edge due to no intersections), it may be unfair to consider this contribution differently from two other images by the same photographer, but located 200 meters apart from each other at different blocks of Oxford Street. Thus, it would be reasonable to perform additional pre-processing of the routing graph and divide edges that are longer than a certain number of meters (e.g. 50, 100 or 200). Deciding what maximum edge length has to be chosen and what needs to be done with short bended segments are the subjects of two additional substantial studies, which could not be conducted within the limits of the PhD research project.

The above two acknowledged issues are not as severe as the problem of edge ends inclusion, which can be confirmed by comparing maps in Figure 5.9 with those in two bottom rows of Figure 5.10. When normalisation is applied to the scores that are based on the standard windows, there exist many instances of 'jumps' between different levels of values. A lot of short segments form local peaks in street attractiveness levels, which may not often happen in reality. When the ends are excluded from the edge windows, spatial distributions of scores become essentially more homogeneous even despite that two other issues are not worked out.

Although the normalised scores are useful for revealing problems in edge windows, they are not believed to be practical in a photo-based routing system. According to the theory in Section 1.2, the gain from choosing an attractive path should become higher with increase of the distance. This requirement is no longer satisfied after the scores are divided by the edge length.

Before considering the effects of edge window boundary blurring and 'vote fission', it was decided to look at how the scores are influenced by the choice of the window size. In addition to 50-meter edge windows (see two top rows in Figure 5.10 on the facing page), scores were calculated for windows of 10, 25, 75 and 100 meters. The results of this experiment for filtered Flickr and Panoramio data are shown in Figures 5.11 and 5.12, respectively.

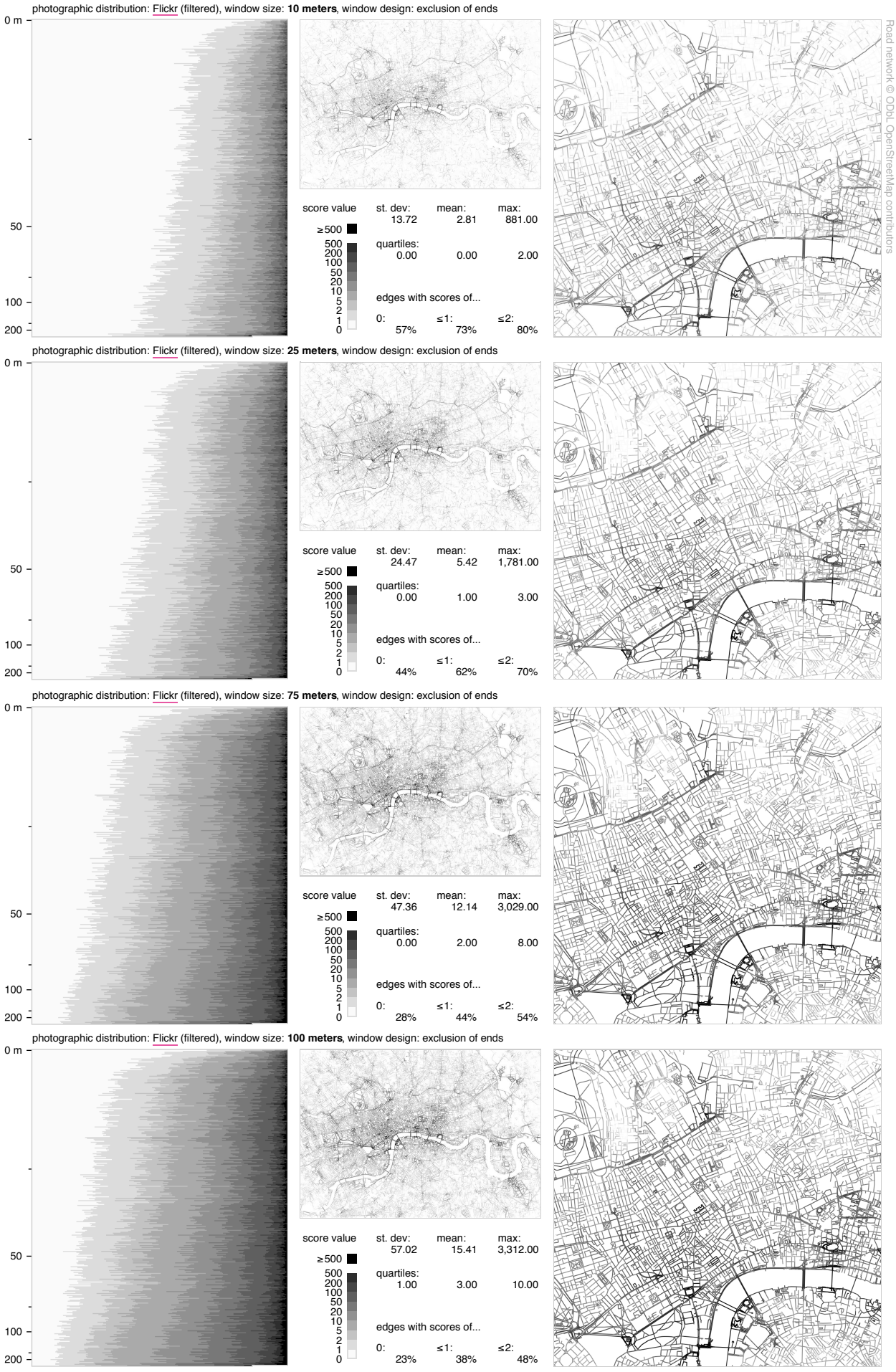


Figure 5.11: Edge windows of four additional sizes (Flickr).

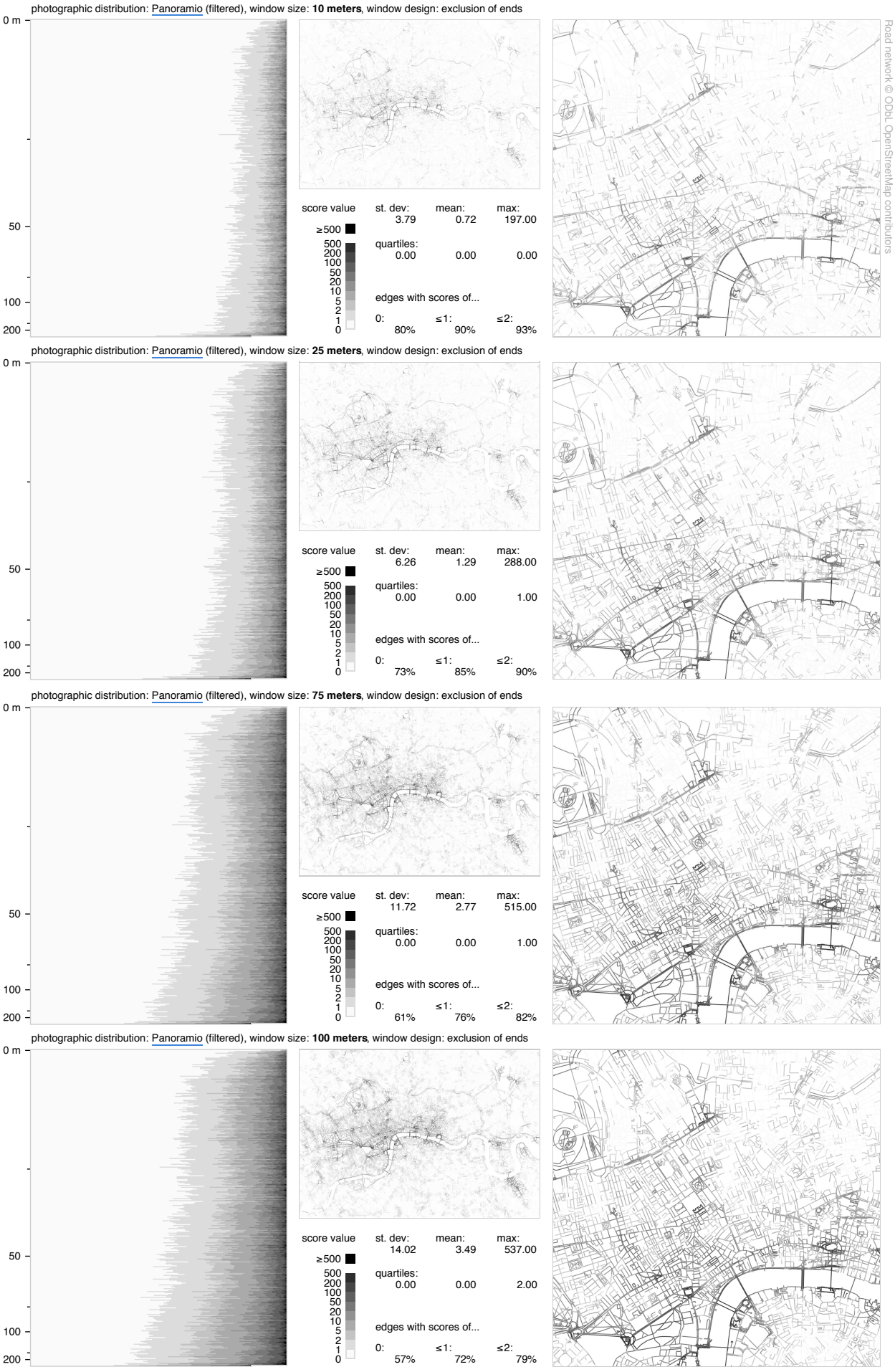


Figure 5.12: Edge windows of four additional sizes (Panoramio).

As expected, the filtered Flickr dataset, which contained about four times more 'votes' than one from Panoramio, produced higher attractiveness scores in all cases. Besides, substantially bigger proportions of edges had non-zero scores. Thus, despite that Flickr was originally further from a *model photographic collection* than Panoramio and remained so after bias reduction (see Figure 4.65 on page 215), there could be a benefit of using this source in a routing system in favour of a cleaner Panoramio data. Importantly, Flickr images have a wider spread away from the city centre, making it possible to distinguish between potentially attractive and non-attractive streets in more urban districts.

The bigger the window size, the higher the mean attractiveness score, so the smaller the proportion of the noise that may be introduced by individual photographers with uncommon behaviour. On the other hand, the smaller the window size, the more specific and 'localised' the scores become. The latter effect can be observed on the maps in Figures 5.11 and 5.12 – the changes in the colours of the roads are smoother with increase of the window size.

Taking into account that Flickr data is richer on 'votes', but the quality of these 'votes' is generally lower, it can be logical to suggest the use of smaller edge windows when dealing with this source of photographs. It may be reasonable to expect that some remaining irrelevant 'votes' such as photographs inside the museums are likely to be distant from the footways. When window size is 75 or 100 meters, it is much more probable that these votes are included into the edge scores and thus add bias to the estimated street attractiveness.

Even when window size is overwhelmingly large, not absolutely all geotagged images form the attractiveness scores, as it is demonstrated in Figure 5.13.

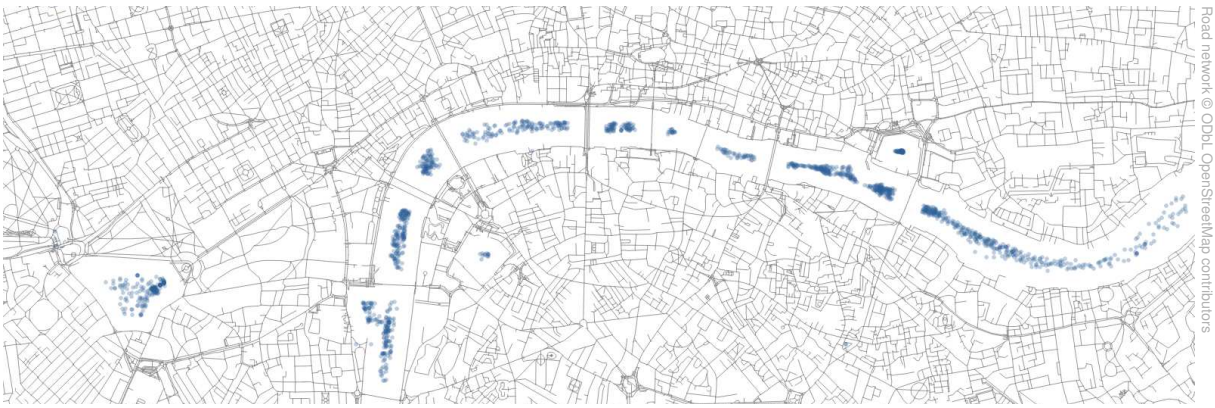


Figure 5.13: Panoramio images that have passed all filters, but have not been included into the largest tested (100 meter wide) windows due to being too far from any of the topology edges.

Experiments have shown that a compromise in a choice of the window size can be found in use of *edge window blurring* (see Figure 3.5c on page 50). With this design, the 'votes' that are closer to a given road segment are translated into larger summands for the overall score, while the images that are geotagged near the boundary of a window are given less 'weight'. As a result, the scores represent a more balanced combination between the attractiveness of *the particular road segment* and *the surrounding area*, and the window does not have to be small.

Edge window blurring can be based on a range of profile functions; the simplest one is linear: $weighted\ vote = 2\frac{window\ size - distance\ from\ vote\ to\ edge}{window\ size}$. Alternatively, this can be a cumulative normal distribution function, arctangent, inverse log, etc. The function can be normalised so that if an edge is located in an area with evenly distributed 'votes', the value of the score is equal to the one that has been obtained with no window blurring ($\int blurring\ function = 1$).

Positive impact on the scores from applying linear blurring was observed for windows of all tested sizes, both for Panoramio and Flickr data. An example is shown in Figure 5.14 below. Comparison maps reveal an interesting change in the values – 'popular' road segments obtain

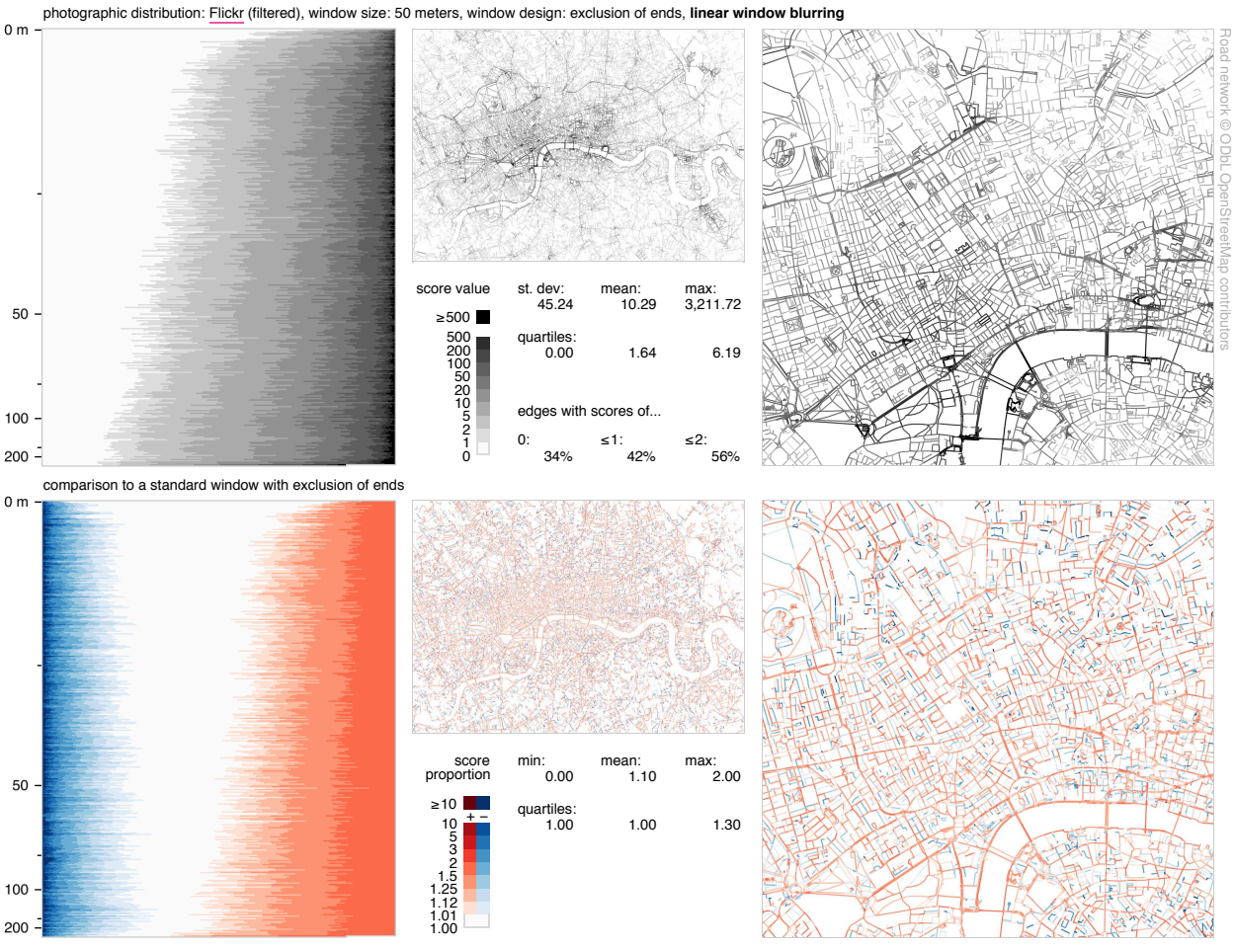


Figure 5.14: Effect of edge window blurring.

even higher scores, while a number of their neighbours become undervalued. The most vivid cases include streets on both sides of a walkway between St. Paul's Cathedral and Millenium Bridge, roads that go in parallel to Whitehall and Regent's canal, and also a number of pathways in Regent's park near the London Zoo. Without window blurring, these network edges would be unreasonably treated as more attractive only because within a few tens of meters from them there exist places where a lot of photographers have taken pictures.

Another potential improvement of edge window design, 'vote fission' (Figure 3.5 on page 50), was not found helpful. According to the original proposal, it could be useful in suppressing the gain from walking past the intersections of roads, i.e. places where geotagged photographs contribute to the scores of several connected edges. Instead, being applied to the real data, this change in the score assignment function introduced a significant amount of unwanted bias.

Urban road network topologies do not always have an even density of edges, and this inequality can be magnified by a non-homogeneous granularity of the map data. Therefore, geographical areas that have complex interlacements of footways and at the same time are represented

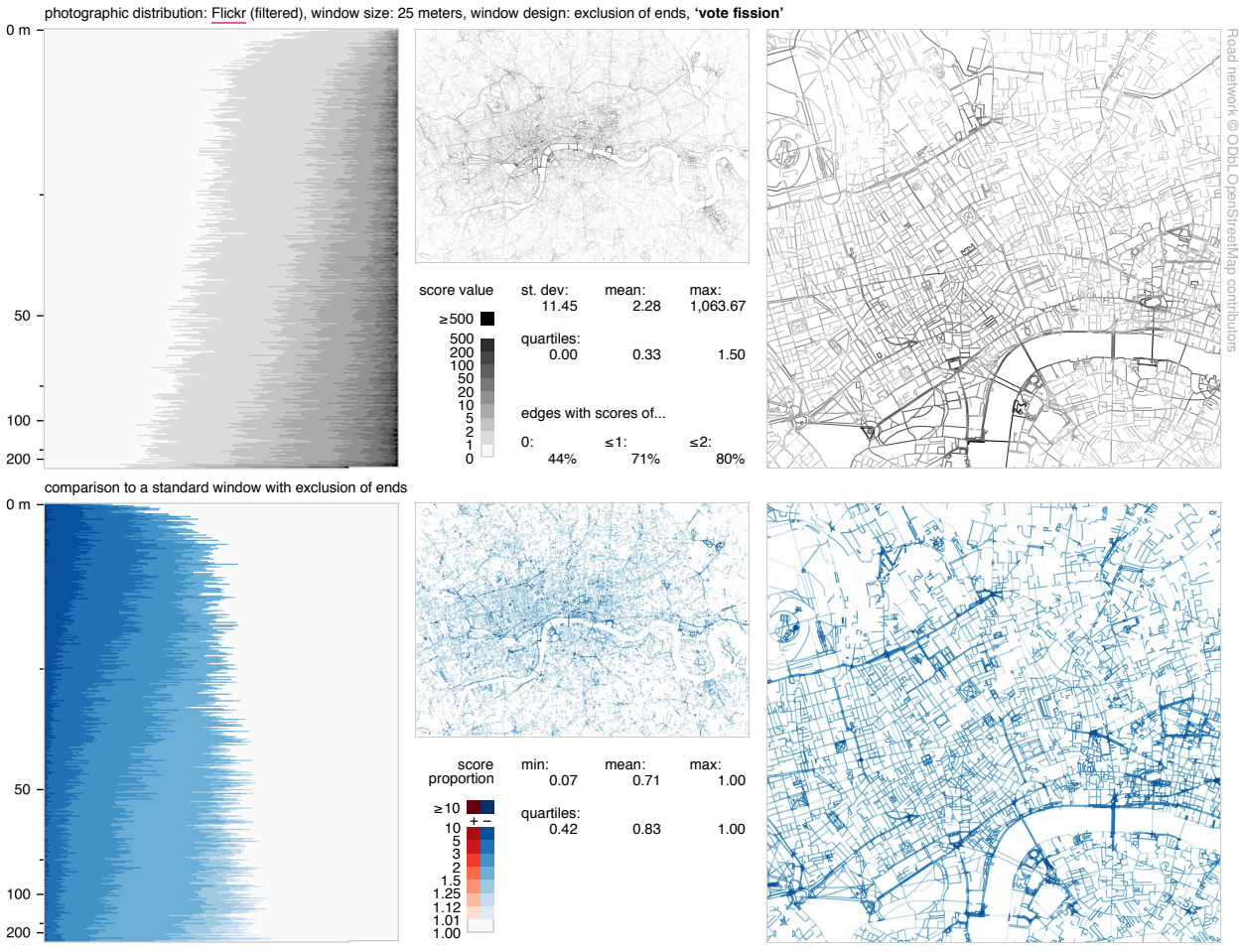


Figure 5.15: Effect of 'vote fission' in edge windows.

with a lot of detail, may not be assigned with high attractiveness scores even when surrounded by a vast number of geotagged photographs. Thus, the scores of the streets with separately mapped pavements are approximately divided by two, and complex footway junctions such as those near the popular landmarks are affected even more extensively. An example of this negative trend is given in Figure 5.15 on the preceding page.

Summarising the above, it can be concluded that the choice of the edge window design is a crucial task for the developers of a photo-based leisure routing system. The decisions that are made at this stage can significantly influence the work of a routing algorithm. Some recommendations such as window blurring or end exclusion may be relevant regardless of the configuration of the network topology or the properties of the photographic data. However, a universal optimal combination for all window parameters can be hardly chosen for any photo-based routing system. For example, it would be unreasonable to use the same window size for Flickr and Panoramio data in London due to the differences in the volume of the 'votes' and their average quality. Similarly, it might be reasonable to tune this parameter for the identical photographic source in another urban area or even in the same place after a lapse of time as the image data coverage may become significantly different. The bigger the number of photographs there are available after filtering and the more widely distributed the geotags are, the smaller the edge window size can be chosen, and therefore the more localised and representative the scores can be made.

5.3.3 Sensitivity to data filtering

Score comparison tests can be useful not only for analysing various edge window designs, but also for assessing the impact of image filtering. If a change in an applied bias-reduction function B (see Equation 1.2 on page 21) affects attractiveness scores of some edges more than of others, it can be a strong sign of the importance of filtering. Bias-reduction function can be tested as a whole and component-by-component, i.e. by removing or adding individual filters.

A number of experiments were conducted as part of this research to see how the computed street attractiveness scores in Central London could be affected by the filters introduced in Chapter 4. Figures 5.16 and 5.17 on the following pages demonstrate the results of two tests for Flickr and Panoramio datasets, respectively.

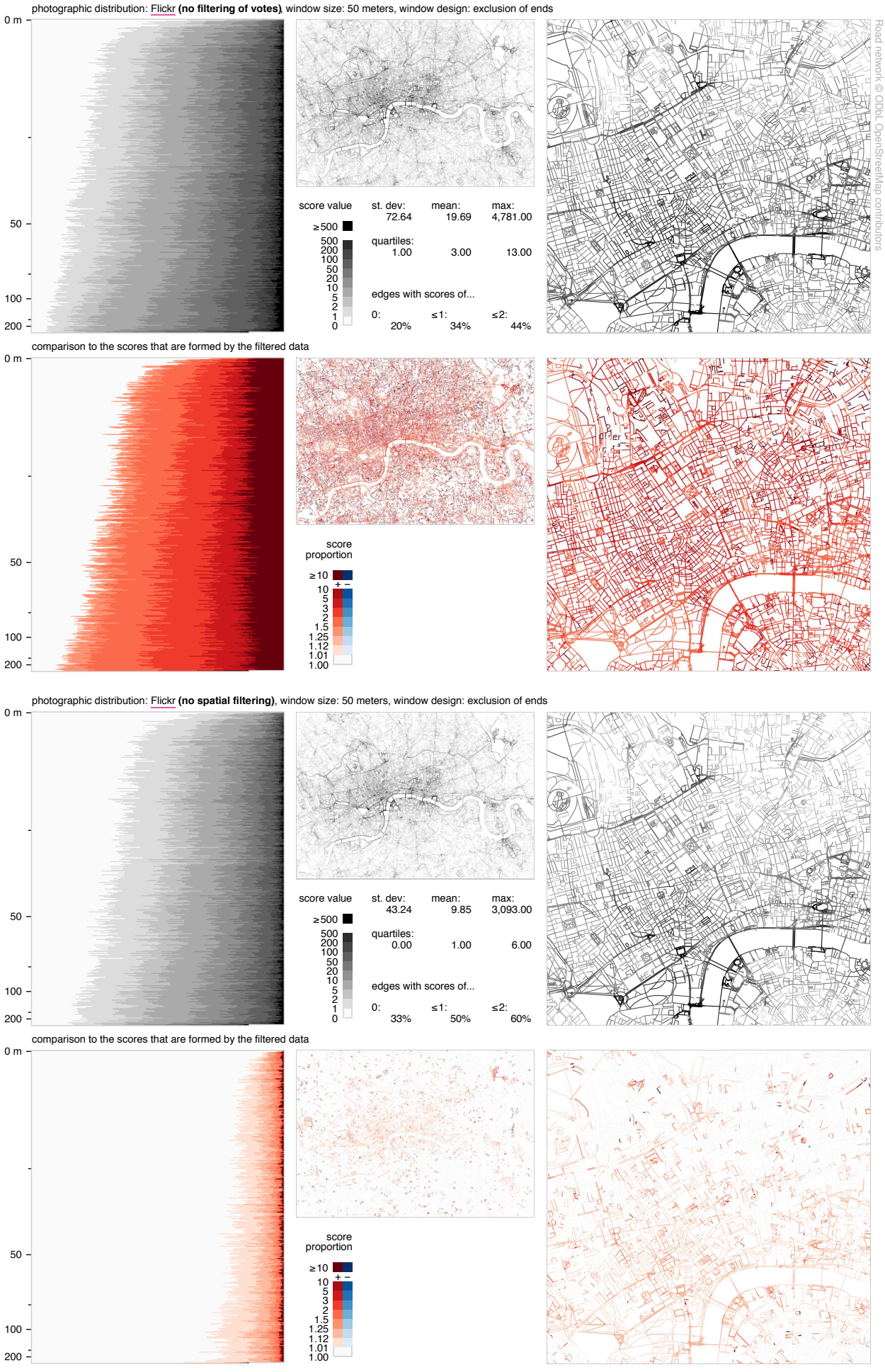


Figure 5.16: Sensitivity of street attractiveness scores to photographic data filtering (Flickr).

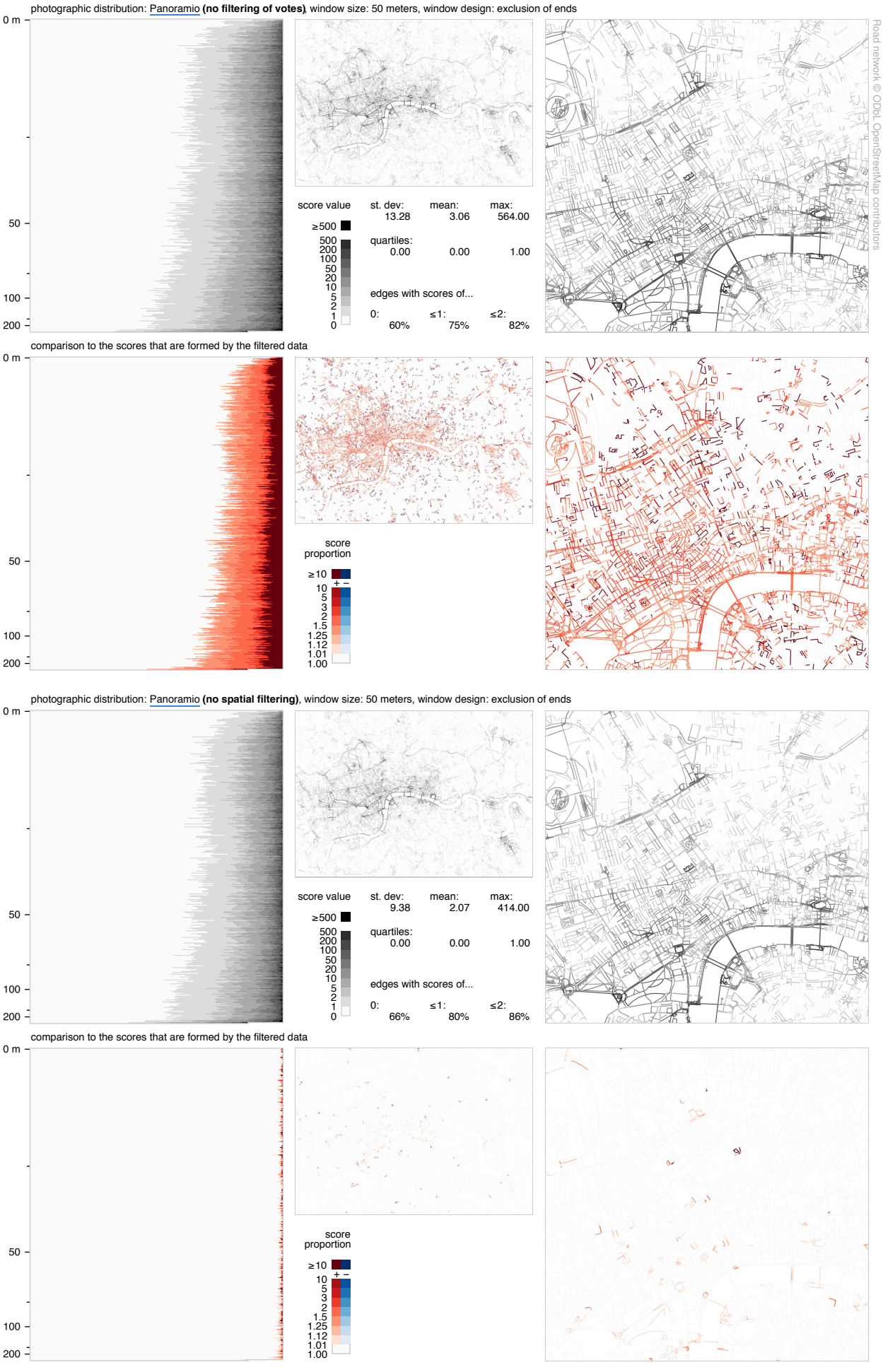


Figure 5.17: Sensitivity of street attractiveness scores to photographic data filtering (Panoramio).

The first test compares edge scores that were obtained using the final filtered version of the data (see Figure 4.65 on page 215) with the scores derived from the initial photographic distributions (Figure 4.18 on page 136). As it could be expected, many edges in the second case have higher attractiveness, as the number of 'rejected votes' in both Flickr and Panoramio photographic collections becomes equal to zero (i.e. bias-reduction function B turns into an *identity function*). The second row of views in Figures 5.16 and 5.17 with the explicit representation of the difference (Gleicher et al. 2011) helps understand the essence of the change.

It can be noted that the proportion between the scores is not even in various geographical locations, and the differences can be observed both at the level of individual road segments and at the level of localities. Perhaps, the first thing that attracts attention is a set of outliers, i.e. the edges that are 10 or more times more 'attractive' in the case of unfiltered data. Many of them are situated away from the city centre, as the smaller maps reveal. The majority of these edges are those that lose all or almost all 'votes' for attractiveness when photo filtering is applied. Such significant changes are likely to occur in rather unpopular areas where the scores are formed by very few images or near the hotspots of *type two* (see Subsection 4.3.3 on page 157). More interesting, however, is a smooth variation of the relative difference. It can be clearly seen that the scores near the parks or by the embankments are less affected by the bias-reduction function than those of the regular streets. This pattern emerges because open areas contain significantly less 'votes' that do not pass the content-based and metadata-based filters (e.g. there are much fewer photographs taken indoors). Oppositely, areas of active social life such as Soho or Camden Town are more affected by the bias-reduction function applied to both Flickr and Panoramio data. The same is true for the neighbourhoods of popular public buildings such as museums or railway stations. The above observations can serve as an essential proof for a need to filter photographic data when they are used in crowd-sourced trip-planning applications or for other similar purposes.

Another example of score sensitivity to the bias-reduction function features a single filter type (see the bottom half of Figures 5.16 and 5.17). It shows how the edge scores change if spatial filtering of photographs is not conducted (the method is justified in Subsection 4.3.3 on page 157). The maps that explicitly represent the difference in the values unveil the locations of hotspots, i.e. spatial coordinates with unreasonably high numbers of photographs form one

or many photographers. The most significant changes occur near the hotspots of *type two* – the points where the 'votes' are likely to be geotagged by means of searching for a place by its name. Apart from that, Flickr-based spatial distribution of edge scores also slightly distorts by the presence of hotspots of *type one* – cases of mass geotagging, which are common among the users of this photo-sharing service. These hotspots are more likely to create noise than bias.

Despite that rather large proportions of photographs are rejected during spatial filtering (see Figure 4.42 on page 175 or Table 4.4 on page 216), it cannot be said that this particular method dramatically influences the measure of street attractiveness. The main reason for this peculiarity is that the scores are formed by the numbers of 'voted' photographers, not the individual images that are located within the edge windows. Thus, if two hundred photographs are placed by some user at the centroid of 'Westminster', they can only increase one or a few scores by about one point. Spatial filtering, however, remains a very important part of the process of bias reduction. First, it does not allow the scores to be influenced by inaccurate geotags, which may result unwanted pathway choices in some situations. Second, it significantly reduces the size of a photographic dataset that needs to be processed. As a result, score-to-edge assignment (i.e. mapping function M in Equation 1.2 on page 21) becomes less resource-intensive.

Being driven by visual analytics, edge score sensitivity tests can be an effective instrument for evaluating various design choices, which influence the values of street attractiveness scores in photo-based leisure routing systems. The examples that have been presented in Sections 5.3.2 and 5.3.3 demonstrate how a combination of linked views together with various comparison techniques can encourage the extraction of the new knowledge and also help confirm a number of previously made conclusions.

5.4 Routing algorithm

The aim of the final part of the project was to implement and to evaluate a sample routing system, which would use the derived street attractiveness scores to suggest attractive leisure walks. This task was solved in line with the methodology that was described in Subsection 3.1.2 on page 49. The routing system received numeric identifiers of starting and ending topology nodes (i.e. road intersections), desired walking time, pace in kilometers per hour, the name of the road topology to work with and the name of the attractiveness score to rely on. First, the algorithm checked the input data and derived the desired route distance (its cost ω'). Second, it searched for the standard shortest path between the given origin and destination and thus checked if the desired walking time was feasible. Then, the algorithm iteratively applied various importance coefficients I_a (see Equation 3.2 on page 54) for the chosen street attractiveness score and invoked the Dijkstra's algorithm to find the most optimal route for the resulting weighted edge costs (the value increased from zero with a step of 0.1). Finally, when a route of a satisfying length was found, the system returned a sequence of edges that were parts of it.

The algorithm was implemented in Java and could be launched via the DAF command line:

```
$ app/console pr:networks:route london_osm_dec2013
    iterative__score__uc_b_e_50__london_panoramio_jul2014__pass_all_filters 29699 316
    --pace 5 --time 120
# london_osm_dec2013: the name of the dataset that belongs to domain "networks" and
    contains the cleaned road topology
# iterative__score__uc_b_e_50__london_panoramio_jul2014__pass_all_filters: algorithm
    type (the same entry point can be used for other approaches to routing) + the name
    of the pre-calculated score (here: user count, linear window blurring, exclusion
    of edges, 50 meters; use the latest version of photographs from Panoramio that
    have passed all introduced filters)
# 29699, 316: identifiers of the origin and the destination (here: Holborn Station and
    Oxford Circus)
# --pace 5 --time 120: pace in kilometers per hour and desired time in minutes
```

The result could be returned to standard output as a wrapped GeoJSON (`http://geojson.org/`) or saved to a `.json` file:

```
{
    "result": {
        "routes": [
            {
                "time": 121,
                "length": 10089,
                "segments": GeoJSON object with all edges in the route
```

```
            }
        ],
        "origin": {
            "nodeId": 29699,
            "lon": -0.1204581,
            "lat": 51.5176085
        },
        "destination": {
            "nodeId": 316,
            "lon": -0.1419463,
            "lat": 51.5152428
        }
    },
    "query": {
        "pace": 5,
        "time": 120,
        "origin": "29699",
        "destination": "316"
        "networkName": "london_osm_dec2013",
        "routerType":
            "iterative__score__uc_b_e_50__london_panoramio_jul2014__pass_all_filters",
    }
}
```


Optionally, it was possible to save a log of all routes, which were derived at each iteration, into a temporary DAF component. This allowed the process of pathfinding to be visualized for debugging or demonstration. In addition, origin and destination could be defined by their spatial coordinates instead of topology node identifiers. In this case the routing system automatically searched for the nearest available nodes and included their coordinates into the response.

Such design allowed the implemented routing system to be easily integrated into a web-based application with a convenient user interface. However, as this task was not included into the scope of the project, all experiments with pathfinding were conducted with use of QGIS (see Subsection 3.3.1 on page 96) and the command line.

The performance of the system was not optimised as this was not critical. All iterations were performed sequentially in a single thread, which made the process of pathfinding rather slow – it took on average between five to thirty seconds to obtain an attractive route. This aspect could be significantly improved if the iterations were parallelised, similarly to how this was done for quad-processing.

This report does not include the individual outcomes of all conducted experiments, but lists some general findings that it was possible to make. An example of a walking route that the system could generate can be found in Figure 5.18 on the next page.

By combining spatial representations of all iteratively generated pathways with the chosen edge attractiveness score and the underlying photographic distribution, it was possible to easily assess the adequacy of the routing algorithm, both including the process and the final result. Visual analytic approach was found suitable for this purpose as it enabled complex multivariable comparisons with no need to introduce convoluted mathematical models. Given that the objective function (the value of gain G from walking along the attractive pathways) had a subjective nature, it was fair to assume that the subjective judgement of the results could be therefore considered as appropriate.

Experiments with various underlying photographic distributions (i.e. filtered and non-filtered versions of Flickr and Panoramio datasets), edge window designs and other parameters of the system were able to support the proposed theory and also emphasised a number of limitations.

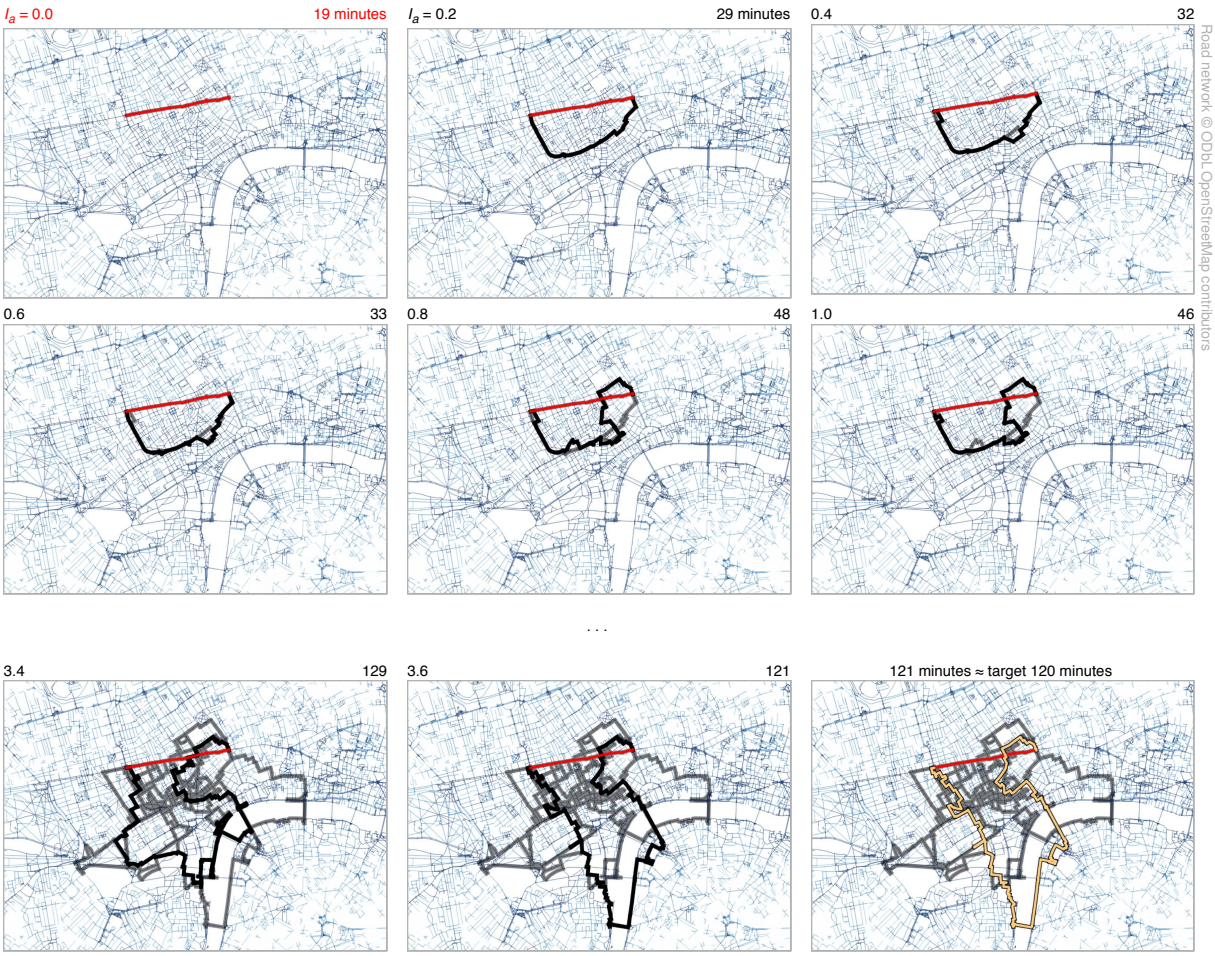


Figure 5.18: Example of a two-hour walking route from Holborn to Oxford Circus that was generated with use of Panoramio data. Pace: 5 $^{km}/_{h}$. The maps show the shortest path between the given points, paths for a number of selected coefficients I_a, paths from earlier iterations and also the resulting path.

One set of tests targeted the differences between filtered Flickr and Panoramio distributions as estimators of the value of street attractiveness. As it could be expected, these two photographic datasets produced various attractive routes given the same user input and other system parameters. Because the overall number of 'votes' in the Panoramio data was less, and their coverage was not as diverse as in Flickr (see Figure 4.65 on page 215), resulting routes more often included the same segments of streets, most of which were located in the very central part of London, approximately within the Congestion Charge zone (Transport for London 2014). Wider spread of Flickr data encouraged the algorithm to give preference not only to the most central roads – it more often produced the routes that included peripheral parts of the town. Lack of data in some areas such as Peckham or Hampstead (especially in Panoramio) made the search for the attractive routes in these places difficult or even impossible – the weighted 'shortest path' between many of the local nodes was not changing with increase of the importance coefficient I_a. This situation raises an important general limitation of the idea of estimating street attractiveness with existing geotagged photographic data – this approach is not applicable in absolutely any location. Thus, if a public-facing leisure routing system has to be operable in as many areas as possible, it may be reasonable to choose a larger source of photographic data even if the average quality of the 'votes' for attractiveness that it contains is lower.

It was found that the overall shape of a leisure route was influenced by the balance between the desired distance and the shortest distance between the origin and destination. When a proportion between these two lengths was high, it was probable for a path to contain remote streets, i.e. those located rather far from both of the given points. An example in Figure 5.18 on the preceding page is an instance of this situation – with the destination being about one and a half kilometers West from the origin, the route extends to the South by three kilometers, being approximately ten kilometers long. This peculiarity makes the routing algorithm potentially sensitive to the shape of the region for which the data are available. If, for example, the road network topology or records from Picasa were only downloaded for a part of London that is North from Piccadilly Circus, the result of a case in Figure 5.18 would be completely different. This observation may suggest the developers of photo-based leisure routing systems to obtain the data for larger areas than they want to cover with their service. Ideally, the size of the introduced gap should be two times smaller than the maximum length of a route that the users may

be allowed to query. In addition, it is also necessary to obtain photographic data for an area that exceeds the region with the network topology by the size of the edge window. Otherwise, the attractiveness scores of the streets that are located at the boundary may be underestimated.

A number of experiments confirmed the sensitivity of generated paths to photographic data filtering. When no bias-reduction function was applied, some routes shifted from parks or embankments to areas with pubs and restaurants or went past railway stations or museums. Filtering by luminance (Subsection 4.5.2 on page 189) was found the most influential among all other bias-reduction methods in Chapter 4.

Ratio R in Equation 3.2 on page 54 (a formula for the combined weighted cost of an edge) was also a subject of a short study. It appeared that smaller values of R in a range between 0 and 1 produced the least flexibility of the routes – they were not largely diverging from the actual shortest path with increase of I_a. Oppositely, when R was approaching its highest allowed value (e.g. was equal to 0.999), the algorithm was able to produce more convoluted routes.

In order to maintain the purity of the experiments, all walkable topology edges were treated equally, i.e. the categories and other properties of the roads were ignored. The shapes of some of the generated routes emphasised the importance of considering more factors in real public-facing photo-based pathfinding systems. The most vivid cases of inaccuracy were the routes that went through the tunnels under the Thames or under a walkway between St. Paul's Cathedral and Millenium bridge. As these road segments are 'indoors', they were not supposed to accumulate any 'votes' by design. This did not happen due to their neighbourhood with attractive places above ground, so some generated routes unreasonably included these road segments. A solution to this particular problem can be simple, if it is known that the considered urban area does not have any tunneled 'attractive streets'. Score-to-edge assignment function can be notified about the status of a currently processed edge and skip it if it is not above ground. In OpenStreetMap, the tunnels are tagged with `tunnel = yes` or `level = -1`.

Apart from the tunnels, pedestrians may be willing to avoid roads with heavy traffic when planning a leisure walk (Davies, Lumsdon and Weston 2012). Therefore, it may be reasonable to suppress the value of gain from choosing primary or trunk roads even if they are surrounded by large number of 'votes' for attractiveness. High scores for these roads may be caused by their proximity to a landmark, which, perhaps, may be also approached by walking along a parallel street.

It is reasonable to suppose that the variability of the attractiveness scores are partially caused by differences in pedestrian traffic, not only by how the humans perceive the urban environment. Therefore, some way of normalising the numbers of 'votes' by the numbers of pedestrians can potentially lead to a better accuracy of the method of estimating street attractiveness with geotagged photographs. This improvement, however, requires data that are extremely hard to collect for all streets in the network (Desyllas et al. 2003).

5.5 System evaluation

A sequence of experiments with photographic data and road network data supported the proposed theory and also helped justify a number of design decisions. Applied research methods, however, did not lead to an answer on the most important question that the developers of the leisure routing algorithms might be interested in – 'Which input data and which parameters of the system do I need to use to generate the most attractive routes for my customers?'

The difficulty of measuring the real effectiveness of a particular solution lies in the *subjectivity* of the gain function $\mathbb{\Gamma}_u(P)$ that the system incorporates (see Section 1.2 on page 17). Indeed, if the goal of the routing algorithm is to suggest a walk that an average user will find more attractive than other options, then it is only possible to estimate the quality of the result by introducing a feedback loop.

The authors of two related research projects (De Choudhury et al. 2010; Quercia, Schifanella and Aiello 2014) conduct quantitative and qualitative user studies to assess their systems. They ask groups of respondents to evaluate the output by leaving Likert-scale rankings or textual feedback. Although the surveys of this kind are able to show that a system is working, they are hardly enabling any improvements, especially if these improvements are hidden behind a change of a single parameter.

To involve the users of a photo-based routing system in its improving, all of them can be turned into the participants of a randomised experiment (Kohavi et al. 2009). For example, an interface of a mobile application that suggests the routes can be supplemented with a simple feedback from, and random groups of users can be dealing with slightly different combinations

of parameters without knowing this. After enough feedback is collected, the least successful configurations can be replaced with the untested ones, moving the system towards a more optional state.

Continuous collection of feedback is crucial if a system needs to be deployed in more than one urban area. The differences in spatial densities of crowd-sourced photographic datasets and also variations in the quality of the resulting 'votes' may make it reasonable to use distinct sources of images or bias-reduction functions in different cities. Besides, some local peculiarities of street networks can be a reason for choosing larger or smaller edge windows.

Mass user testing of the implemented routing algorithm was not included into the scope of this research, as this would significantly exceed its budget. However, some evaluation of the system 'on the ground' could be still made by means of several trial walks along the generated routes. With use of the Dataset Abstraction Framework (Section 3.2), it was possible to easily set up the routing system outside London, which extended the geography of tests to Birmingham, York and Newcastle (see an example in Figure 5.19 on the next page). The findings gleaned from this experience cannot pretend on a complete trustworthiness, however, it can be fairly said that this cheap method of algorithm evaluation may be rather useful for some primary system adjustment. Similarly to the software usability tests where as little as five users may reveal around 85% of all problems in an interface (Nielsen 2000), a few trial walks by a small group of people may produce a comprehensive list of improvement suggestions. The following observations, which the author of this work made during his trial walks at the end of the project, can be worth sharing with the developers of future photo-based routing systems:

> It appears that considering the categories of roads in addition to the attractiveness scores can indeed lead to a significant improvement of the walking experience. This issue, discussed in the previous section on page 250, was confirmed during the trial walks.
>
> Inclines or presence of stairs may also contribute to the walking experience, especially for certain categories of users. It can be recommended to consider this factor when calculating weighted costs for the routes. The issue of accessibility can be especially topical in hilly areas such as North London or central parts of Newcastle or Birmingham.

Another issue that relates more to the quality of the underlying road topology rather than to the used photographic data is about the locations of intersections with car traffic along the chosen route. Particular road junctions may be lacking crossings at some of their sides, so the users of a routing system have to make local detours. This action increases the overall walking time and may be subjectively perceived as inconvenient.

In few rare cases it may be beneficial to involve a new 'vote'-filtering method, not introduced in Chapter 4. It has been found that some open-air places with restricted or limited pedestrian access can unexpectedly increase the scores for the surrounding streets, even when the edge windows are relatively small. Examples include Buckingham Palace Gardens, Tower of London and St James' Park Stadium in Newcastle. Because the photographs that have been taken within these territories cannot be easily marked as irrelevant based on their metadata or content, it can be tried to remove them with use of

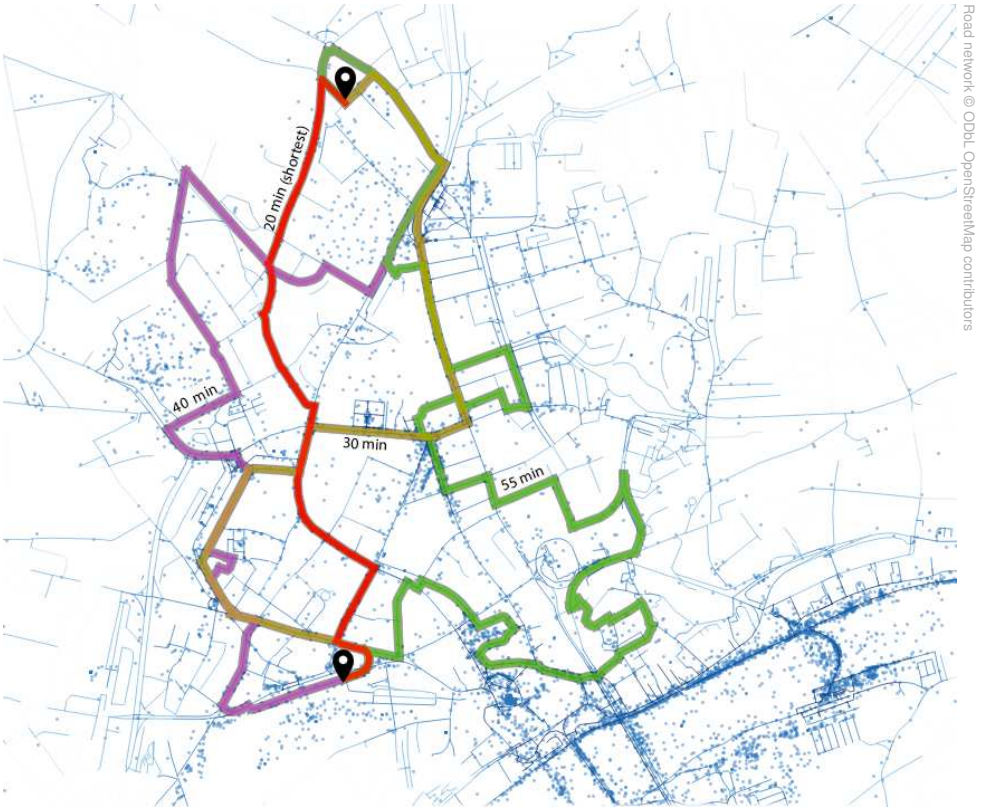


Figure 5.19: System evaluation in Newcastle. The map shows four walking options from Old Library Building (Newcastle University) to the Central Railway station. Network scores are based on Panoramio and OpenStreetMap data that were collected in January 2014.
`https://twitter.com/kachkaev/status/420908952105148419`

polygonal masks. With access to cartographic databases such as OpenStreetMap, it is technically possible to obtain a collection of local private gardens, stadiums or other restricted territories to then reject all the photographic records that are contained within them. This filtering method may need to be adjusted with respect to the average spatial accuracy of the polygons as well as the photographic records. Used together with edge window blurring (see page 238), photo distribution masks can potentially reduce the occurrences of errors in the street attractiveness scores.

To summarise, this chapter demonstrates how interactive visualization of the data for a photo-based leisure routing system together with a small number of trial walks can significantly reduce the search space for an optimal configuration of a pathfinding algorithm.

Chapter 6

Conclusions

This chapter summarises the findings of this research and contains the discussion of its results.

6.1 Revisiting specific objectives

The overall goal of this project was to study the potential of crowd-sourced photographic collections in estimating street attractiveness and to examine the ways of utilising this information in automated route planners for pedestrians. The problem of generating attractive leisure walks based on spatial densities of geotagged images was introduced in Section 1.2, which lead to the formulation of research aims and objectives (Section 1.3). This section assessed the degree to which the objectives have been satisfied by revisiting the research questions defined in Section 1.4.

RQ 1: What sources of user-generated photographic data are suitable for automatic creation of attractive pedestrian routes?

The answer to this question consists of two parts – theoretical and empirical. First, this work introduces a concept of a *model photographic collection* and defines its parameters in Subsection 3.1.1. This becomes a part of a theoretical framework, which can be utilised regardless of the source of the actual data. Second (in Chapter 4), the research examines four collections of images (Flickr, Geograph, Panoramio and Picasa)

in a single geographical region (Subsection 3.1.3) and assesses them in accordance with the proposed theory. This part of the study focuses on revealing patterns that make the chosen datasets incompatible with a definition of a *model photographic collection*. The analysis is done by means of the interactive spatiotemporal exploration of the harvested photographic records (Section 4.3) and with use of an online public survey (Section 4.4). The obtained new knowledge about the data suggests that among the four chosen sources Pareto-efficient are Flickr and Panoramio (see Section 4.6).

RQ 2: What features of geotagged photographs can be used in pathfinding?

Taking into account the fact that the data in the photo-sharing services are not designated for their use in the trip-planning applications, this research questions a problem of weighting various user entries differently in order to reduce a potential bias in the result. Subsection 3.1.1 suggests a general methodology for binary filtering of contributions based on their spatiotemporal coordinates, metadata and content. A summary of all photographic features available for the chosen sources can be found in Table 4.1 on page 118.

RQ 3: What methods should be applied for data filtering in order to remove unwanted user entries?

First, this question is addressed theoretically in Subsection 3.1.1, where it is suggested to start with collection-level filtering by spatial and temporal coordinates, continue with filtering by additional metadata and finish with content analysis if necessary. It is suggested that adhering to this sequence of steps may be the least resource-intensive when dealing with real data. Chapter 4 introduces a set of specific techniques that can lead to less bias in the resulting measure of street attractiveness. These techniques are either informed by the discovered anomalies in spatiotemporal distributions of images or by the subjective opinion of respondents, who took part in the photo content assessment survey (`http://www.photoassessment.org/`). The results of the survey have not only helped better know the differences between the sources of photographic data, but have been also involved in the adjustment of filters. Some of the proposed bias-reduction methods have been found ineffective for the given collections of data and therefore have not been applied. The result of all experiments with the chosen crowd-sourced photographic collections is summarised in Table 4.4 on page 216.

RQ 4: How to obtain the attractiveness scores of the road network edges in a routing graph?

The problem of converting point-based spatial distributions of photographs into street attractiveness scores is discussed in general terms in Subsection 3.1.2. Here, a concept of *edge window* is introduced and different potential designs of these windows are suggested. Chapter 5 focuses on the practical application of the theory – the process of *score-to-edge assignment* is studied in Section 5.3. First, an efficient algorithm for this particular task is proposed and described; it allows considerable computational resources to be saved by introducing a concept of *photo window* and splitting the process into two stages. Second, a number of sensitivity tests examine how the scores are affected by the changes in the photographic data or the form of the edge window. This helps understand which edge window designs are more suitable for the task and also highlights the importance of bias reduction in the real crowd-sourced photographic data.

RQ 5: How to implement the routing algorithm?

The problem of finding a route that can be characterised by a high gain and a fixed cost is discussed in Subsection 3.1.2. In this work, the task is approached as a weighted SP problem, where one of the factors, being formed by the derived edge attractiveness scores, is negative. The implemented software solution is described in Section 5.4. A set of experiments with the resulting routing system conclude the empirical data analysis and allow more conclusions and findings to be made.

RQ 6: How to asses the results?

A methodology that could be used in future photo-based leisure routing systems for maintaining high quality of the automatically generated paths is discussed in Section 5.5. Being rather difficult to be incorporated into this research project due to the limited budget, methodology testing is not included into the scope. Alternative method of system evaluation – *trial walks* – is proposed as a cost-effective yet productive alternative for revealing the most significant problems of a particular solution. Several trial walks, which the author of this work took at the end of the project, supplement the report with a few findings that may be considered useful by the developers of the similar systems in future.

6.2 Research outcomes

The outcomes of this research can be split into two main categories: those that are relevant to the problem under study and those that are potentially applicable in a wider range of tasks. This section summarises both of them.

6.2.1 Contribution to the relevant field of research

Unlike the studies that focus on a narrowly defined problem and search to the bottom of it, this work considers a rather broad and complex topic and is not aiming to provide an ultimate optimal solution. The idea of linking crowd-sourced photographic content with pathfinding involves a wide range of questions from understanding human behaviour to information processing. Therefore it cannot be fully embraced at once and needs to be approached step-by-step from different angles.

Taking into account the findings of various adjacent projects, this research broadens the knowledge about the data available in photo-sharing services and further explores the ability of these data to describe the urban environment. Consequently, collections of crowd-sourced geotagged images may become better predictors of street attractiveness and are more likely to be successfully used in various leisure routing systems. Apart from that, this work examines a problem of converting point-based spatial distributions of data records into numeric scores that predict the attractiveness of individual streets in a road network. New insights gleaned at this stage of the research promote a better understanding of various aspects of the nature of this process. Finally, a sample routing system, which has been built towards the end of the project, helps evaluate the effectiveness of various design decisions and demonstrates the overall capability of the idea of using crowd-sourced images in pathfinding.

The contribution of this work to the field of linking geotagged photographic data with journey planning can be split into several logical units. Each of them has enough self-sufficiency to be independently reused by others in future.

Theoretical framework

This report begins with defining a problem of planning leisure walks, which unlike functional walks are characterised with a gain function in addition to a cost function. Narrowing the problem down to finding routes through attractive streets given a fixed time budget, Section 1.2 suggests a formal approach to utilising crowd-sourced photographic data for knowing which streets might be more preferable for their inclusion in a walk. Following this introduction, Chapter 3 and in particular Section 3.1 presents a general structure of a photo-based leisure routing system and specifies the functionality of its components. Subsection 3.1.1 describes a concept of a *model photographic collection* by expanding previously made assumptions into a set of requirements that a collection of photographs must satisfy in order to be a reliable estimator of street attractiveness. Next, the idea of *bias reduction* in real photographic data is discussed. This leads to a workflow that can be used for analysing arbitrary collections of geotagged images against the proposed requirements and for filtering irrelevant entries. Subsection 3.1.2 looks at the problem of linking cleaned photographic datasets with the street network data. It considers various methods of mapping spatial densities of images into street attractiveness scores and then suggests a possible solution to the defined pathfinding problem.

Detached from any particular collections of data, the above steps form a general task-specific theoretical framework, which can be reused in future leisure routing systems or be adopted for similar problems. The suggested research methodology is followed in Chapters 4 and 5.

New empirical knowledge of four crowd-sourced photographic datasets

Chapter 4 focuses on the analysis of geotagged photographic data in Central London, a region that was chosen in Subsection 3.1.3 for the experimental part of this research. Four sources of images are included into the empirical study: Flickr, Geograph, Panoramio and Picasa. With use of visual analytics, statistical analysis and an online survey, this research confirms previously known patterns in the data and reveals a number of the new ones. Importantly, the chosen method for analysis helps emphasise the differences between the given sources of photographs and know more about the behaviour of users that have formed the datasets.

The knowledge gleaned at this part of the project made it possible to see how distant each of the considered datasets is from a *model photographic collection*. Most of the findings may be of interest not only to those who want to combine photographic data with journey planning.

Methods of bias reduction in photographic datasets as estimators of street attractiveness
Informed by the discovered discrepancies between the real datasets and a *model photographic collection*, Chapter 4 suggests specific filtering methods, which can be applied to remove potential bias in the derived street attractiveness scores. The methods are classified into three groups. The first group (Section 4.3) considers spatial and temporal attributes of the photographs and targets anomalies in their distributions caused by global events, images with inaccurate coordinates (*hotspots*) and local events. A detailed study of individual user behaviour concludes that personal bias can be hardly removed by distinguishing between 'too much active' and 'normal' users. Instead, it is proposed to calculate street attractiveness scores by counting photographers, not the individual images that are nearby a street. The second and the third group of filters (metadata-based and content-based) are discussed together in Section 4.5. They are suggested and adjusted with use of the samples of images, which have been classified by participants of a photo content assessment survey (Section 4.4). The most powerful method of bias reduction among all appears to be filtering by EXIF tags (Subsection 4.5.2).

Although all proposed filtering methods were applied only to data in Central London, they are expected to be appropriate in other parts of the world. Thus, the developers of photo-based routing systems or trip planners can utilise the findings of this research to improve the reliability of their systems. The importance of image data filtering is demonstrated in Subsection 5.3.3.

Understanding how attractiveness scores can be mapped to streets
Experiments with the road network data from OpenStreetMap and the photographic data from Flickr and Panoramio reveal strengths and weaknesses of various proposed attractiveness mapping functions (*edge window designs*). This knowledge may be of direct benefit for those who want to build a leisure routing system that uses densities of crowd-sourced photographs for measuring street attractiveness. An empirical study of differences in the scores with the changes in the approach to mapping is described in Subsection 5.3.2. Its results can reduce the effort to finding appropriate combinations of parameters in future real-world solutions.

The target audience of this work may also be interested in the applied technical solution for data mapping. Two-step approach for score-to-edge assignment, which is described in Subsection 5.3.1, can save a substantial amount of resources, especially if a routing system has a wide geographical coverage or is a subject to fine adjustments.

A complete prototype of a photo-based routing system for suggesting leisure walks
Being the final outcome of this research, the designed sample photo-based routing system can be a source of inspiration for the developers of various web services and also generate more interest to a relatively new opportunity of linking crowd-sourced data with travel planning. The adopted approach to route finding is explained in Subsection 3.1.2 and its implementation is described in Section 5.4. The fact that the walks are generated by the system solely with use of photographic data and with no respect to other features of the environment suggests that not all of them may be perceived by potential users as good choices. Previous research (Davies, Lumsdon and Weston 2012) and a number of trial walks (Section 5.5) raise an importance of combining street attractiveness scores with other data even when there are no doubts in their reliability and the system is finely tuned. This concludes that the knowledge about a street network that may be derived from geotagged photographic collections is more helpful as *one of the factors* in a multivariate walk planner rather than on its own.

An interesting peculiarity of a photo-based leisure routing system is that both its input and output have a subjective nature. This fact makes it difficult to rely only on traditional research methods such as statistical analysis to find an optimal link between them – this would require a model of an extremely high complexity in order to take into account all existing phenomena. This project extensively uses visual analytics to answer most of the research questions and to derive the majority of the findings; justification of this choice can be found in Section 3.3. Although the results of data analysis with VA may be slightly inaccurate due to cognitive limitations and bias, this approach can be recommended to those who plan to conduct similar research. Numerous visual representations, layouts and interfaces that have been designed for this project add to its contribution to the chosen field of study and are free to be utilised for solving adjacent problems.

6.2.2 By-products

Glyph-based survey data analysis tool
One of the steps of studying crowd-sourced collections of images was an online survey, the aim of which was to subjectively classify a sample of nine hundred photographs by seven

criteria. The survey recruited 359 volunteers who left 8,434 responses (49,285 answers to questions in total). Being interested in a detailed examination of the collected data, this research suggests a novel glyph-based approach to their exploration. The method is described Subsection 3.3.4 and Subsection 4.4.2. Used as a visualization technique in a custom designed interactive interface that accommodates multiple linked views, glyphs facilitate pattern extraction in raw survey data and thus lead to better understanding of participants' behaviour as well as features of the subjects. Glyphs can be utilised for the analysis of similar collections of survey responses, which has been shown in Kachkaev, Wood and Dykes (2014). The dataset with all classifications is freely available and may be reused in other studies: `http://github.com/kachkaev/survey-glyphs`.

Empirical knowledge about the limitations of photo service APIs

An important requirement to crowd-sourced photographic datasets when they are to be involved in route planning is the representativeness of their spatial distributions. Therefore, the process of data harvesting plays an important role in the reliability of the resulting system. Section 4.2 contains a detailed investigation of a spatial search function in the APIs of three photographic services: Flickr, Panoramio and Picasa. The result of this investigation is a summary of limitations and pitfalls, which is supplemented with the methods of their avoidance. This knowledge may be of interest to those who plan using crowd-sourced geotagged collections of photographs in research or software products.

Dataset Abstraction Framework (DAF)

The third by-product of the thesis a methodology for processing complex collections of data under the conditions of uncertainty. This question is addressed in Section 3.2. The result of considering various aspects of this generalised problem is a software framework, which has been used throughout the experimental part of this research. Having a number of additional layers of data abstraction, a single control point, a modular structure and other features, the framework makes it easy to organise data analysis in situations when the data come from various sources, don't have persistent structure and contain non-trivial dependencies. Framework usage example is described in Subsection 3.2.5.

Example case study in visual analytics

The final and the most widely applicable outcome of this research is its contribution to the field of visual analytics. The underlying topic can be considered as an example of a problem that requires understanding of complex yet unstructured collections of data. Some of the new visual designs that were proposed and utilised in Chapters 4 and 5 may be re-used in other research tasks, which are not necessary related to photographic data or road networks.

6.3 Future work

Being a step towards a solution rather than a final solution, this research not only answers the questions that existed before its beginning, but also generates a variety of new ones. There is scope for future research to improve some of the proposed image filtering methods, suggest and test the new ones, enhance the process of score-to-edge mapping and revise the approach to pathfinding. The experiments that were conducted in Central London can be repeated in other regions of the world or in the same place after a few years.

Some ideas for system improvement are mentioned Chapters 4 and 5. For example, the effect of seasonal bias in crowd-sourced photographic data can be empirically studied in cities with a continental climate, and a general approach to a relevant bias-reduction can be proposed. The issue of seasonal changes in photographers' activity has been recognised by Andrienko et al. (2009) and Alivand and Hochmair (2013), however still lacks a detailed investigation. Some spatiotemporal filters that have been used in this research can be improved by the development of a formal approach to thresholding. Thus, distinguishing the events and hotspots in photographic datasets of any volume and purpose can be made automatic with no need to involve visual analytics to verify the rules that need to be used. Metadata-based and content-based filters can benefit from new ways of their verification; there is also scope for proposals and adjustments.

Unlike a few projects that are mentioned in Chapter 2 as considering the problem of trip personalisation, this work is not trying to customise the attractiveness scores for people with various interests. However, the analysis of the photographers' behaviour in Subsection 4.3.1

and the results of previous studies make the customisation of leisure routes a promising scope for a new research. This idea may be more practical in several years when photo sharing gains additional popularity and the crowd-sourced photographic datasets grow in size. Because route customisation implies extensive filtering of 'votes' for street attractiveness, an accompanying problem of increasing personal bias should be taken into account. As opposed to route customisation, it can be attempted to merge photographic collections from several sources in order to obtain a larger combined dataset with a possibly better spatial coverage. In this case, it may be reasonable to consider detection of duplicate photographs or even identify duplicate user accounts that the same photographers may have on different photo-sharing websites.

The problem of mapping point-based spatial distributions of photographs into numeric scores for street segments can be also studied further. For example, it would be interesting to know the scores can be improved as a result of splitting long topology edges into segments. More designs and sizes for edge windows may be introduced and tested. A system that could automatically suggest the most optimal form of the windows for a given street topology and a photographic dataset would significantly simplify the deployment of photo-based routing systems in new geographical areas.

Finally, a lot more can be done with the routing as such. The problem of finding a path that has a high amount of gain given a fixed budget (Equation 1.3 on page 22) can be approached not only as a weighted shortest-path problem, but also in a variety of other ways. Another promising direction of research consists in combining derived attractiveness scores with other data to better reflect the real demand of pedestrians. Ultimately, the results of this work can be integrated into a multimodal transportation system where planning of attractive leisure walks is only one option out of many.

If photo-based routing systems become popular in future, this may lead to a radically new challenge related to bias-reduction in street attractiveness scores. Being aware of the principles that are involved in pathfinding at some online service, owners of shops, cafes and restaurants might want to influence the algorithm in order to raise the traffic of potential customers. By taking and sharing geotagged photographs near their businesses using multiple user accounts, they will fraudulently increase attractiveness scores of the surrounding streets and thus make the discussed approach to street assessment less reliable. This potential method of fraud can

even create a new type of underground business similar to those that sell artificial Twitter followers (Stringhini et al. 2012) or 'likes' on Facebook (De Cristofaro et al. 2014). Therefore, the developers of the photo-based routing services can be advised not to disclosure the source of data they rely on and also be ready to handle the issue of deliberately fabricated 'votes'.

The author of this work believes that automated planning of leisure walks where attractive streets get preference can be found useful by many people, especially among tourists who are not very familiar with a region of their interest. Increased informativeness and reliability of a journey planner for pedestrians may make walking a more inviting and enjoyable activity.

Bibliography

Abramovich, S. and Sugden, S. 2005. "Spreadsheet conditional formatting: An untapped resource for mathematics education." *Spreadsheets in Education (eJSiE)* 1(2):3

Aho, A., Hopcroft, J. and Ullman, J. 1974. *The design and analysis of computer algorithms*. Addison-Wesley series in computer science and information processing Addison-Wesley Pub. Co.

Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X. and Saltz, J. 2013. "Hadoop GIS: a high performance spatial data warehousing system over mapreduce." *Proceedings of the VLDB Endowment* 6(11):1009–1020
`http://dl.acm.org/citation.cfm?id=2536227`

Alexa 2014. "Top Sites by Category: Computers/Internet/On the Web/Web Applications/Photo Sharing."
`http://bit.ly/alexa_top_photo-sharing_websites`

Alivand, M. and Hochmair, H. 2013. "Extracting scenic routes from VGI data sources.". ACM Press pp. 23–30
`http://dl.acm.org/citation.cfm?doid=2534732.2534743`

Altshuller, G. 1999. *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical Innovation Center, Inc.

Ambler, S. 2014. "Mapping Objects to Relational Databases: O/R Mapping In Detail."
`http://www.agiledata.org/essays/mappingObjects.html`

Amegashie, J. 2007. "Intentions, Insincerity, and Prosocial Behavior." http://papers.ssrn.com/sol3/papers.cfm?abstract_id=986232

Ames, M. and Naaman, M. 2007. "Why we tag: motivations for annotation in mobile and online media." In *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 971–980

Anca-Livia, R., Stöttinger, J., Ionescu, B., Menéndez, M. and Giunchiglia, F. 2012. "Representativeness and Diversity in Photos via Crowd-Sourced Media Analysis." http://eprints.biblio.unitn.it/4000/

Andrienko, G., Andrienko, N., Bak, P., Kisilevich, S. and Keim, D. 2009. "Analysis of community-contributed space- and time-referenced data (example of Panoramio photos)." In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. pp. 540–541

Andrienko, G., Andrienko, N., Mladenov, M., Mock, M. and Pölitz, C. 2010. "Discovering bits of place histories from people's activity traces." In *IEEE Symposium on Visual Analytics Science and Technology*. pp. 59–66

Andrienko, G., Andrienko, N., Mladenov, M., Mock, M. and Politz, C. 2012. "Identifying Place Histories from Activity Traces with an Eye to Parameter Impact." *IEEE Transactions on Visualization and Computer Graphics* 18(5):675–688 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6018964

Antoniou, V. 2011. "User generated spatial content: an analysis of the phenomenon and its challenges for mapping agencies" PhD thesis, University College London.

Antoniou, V., Morley, J. and Haklay, M. 2010. "Web 2.0 geotagged photos: Assessing the spatial dimension of the phenomenon." *Geomatica* 64(1):99–110

Apple 2013. "Core Image Programming Guide." http://apple.co/1U245Cd

Apple 2014. "iOS 7: Understanding Location Services." http://support.apple.com/kb/ht5594

Arase, Y., Xie, X., Hara, T. and Nishio, S. 2010. "Mining people's trips from large scale geo-tagged photos." In *Proceedings of the international conference on Multimedia*. pp. 133–142

Baeza-Yates, R. 2009. "User generated content: how good is it?" In *Proceedings of the 3rd workshop on Information credibility on the web*. pp. 1–2

BBC 2009*a*. "Heavy snow hits much of Britain." `http://news.bbc.co.uk/1/hi/uk/7864395.stm`

BBC 2009*b*. "Live: G20 Summit Build-up." `http://news.bbc.co.uk/1/hi/business/7973178.stm`

Beeharee, A. and Steed, A. 2006. "A natural wayfinding exploiting photos in pedestrian navigation systems." In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. pp. 81–88 `http://dl.acm.org/citation.cfm?id=1152233`

Borgo, R., Kehrer, J., Chung, D., Maguire, E., Laramee, R., Hauser, H., Ward, M. and Chen, M. 2012. "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications." In *Eurographics 2013-State of the Art Reports*. The Eurographics Association pp. 39–63

Boutell, M. and Luo, J. 2004. "Photo classification by integrating image content and camera metadata." In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 4 pp. 901–904

Brandes, U. and Nick, B. 2011. "Asymmetric relations in longitudinal social networks." *Visualization and Computer Graphics, IEEE Transactions on* 17(12):2283–2290 `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6064994`

Briney, A. 2008. "Geodetic Datum - WGS 84 - NAD 83 - GPS." `http://geography.about.com/od/geographyintern/a/datums.htm`

Camera & Imaging Products Association 2010. "Exchangeable image file format for digital still cameras: Exif Version 2.3."

Cawthon, N. and Moere, A. 2007. "The effect of aesthetic on the usability of data visualization." In *Information Visualization, 2007. IV'07. 11th International Conference*. pp. 637–648
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4272047

Chinchuluun, A., P. Pardalos, A. Migdalas, L. Pitsoulis and P. Pardalos, eds 2008. *Pareto Optimality, Game Theory And Equilibria*. Vol. 17 of *Springer Optimization and Its Applications* New York, NY: Springer New York.
http://link.springer.com/10.1007/978-0-387-77247-9

Cohen, J. 2008. "What is the difference between a framework and a library?"
http://stackoverflow.com/questions/148747/

Crampton, J. 2002. "Interactivity types in geographic visualization." *Cartography and geographic information science* 29(2):85–98

Crandall, D., Backstrom, L., Huttenlocher, D. and Kleinberg, J. 2009. "Mapping the World's Photos." In *Proceedings of the 18th international conference on World wide web*. WWW '09 New York, NY, USA: ACM pp. 761–770
http://doi.acm.org/10.1145/1526709.1526812

Davies, N., Lumsdon, L. and Weston, R. 2012. "Developing Recreational Trails: Motivations for Recreational Walking." *Tourism Planning & Development* 9(1):77–88
http://www.tandfonline.com/doi/abs/10.1080/21568316.2012.653480

DB-Engines 2014. "Ranking - popularity ranking of database management systems."
http://db-engines.com/en/ranking

De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R. and Yu, C. 2010. "Automatic construction of travel itineraries using social breadcrumbs." In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. pp. 35–44

De Cristofaro, E., Friedman, A., Jourjon, G., Kaafar, M. and Shafiq, M. 2014. "Paying for Likes?: Understanding Facebook Like Fraud Using Honeypots.". ACM Press pp. 129–136
http://dl.acm.org/citation.cfm?doid=2663716.2663729

Degtyarev, N. and Seredin, O. 2010. "Comparative testing of face detection algorithms." In *Image and Signal Processing*. Springer pp. 200–209.
http://link.springer.com/chapter/10.1007/978-3-642-13681-8_24

Department for Transport 2011. "Local Transport Note 1/11: Shared Space."
http://assets.dft.gov.uk/publications/ltn-01-11/ltn-1-11.pdf

Desyllas, J., Duxbury, E., Ward, J. and Hudson-Smith, A. 2003. "Pedestrian demand modelling of large cities: an applied example from London."

Deutskens, E., De Ruyter, K., Wetzels, M. and Oosterveld, P. 2004. "Response rate and response quality of internet-based surveys: an experimental study." *Marketing letters* 15(1):21–36
http://link.springer.com/article/10.1023/B:MARK.0000021968.86465.00

Dijkstra, E. 1959. "A note on two problems in connexion with graphs." *Numerische Mathematik* 1(1):269–271
http://link.springer.com/10.1007/BF01386390

Dobias, M. 2014. "QGIS Application – Feature request #3200: cache."
https://hub.qgis.org/issues/3200

Douglas, D. and Peucker, T. 1973. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." *Cartographica: The International Journal for Geographic Information and Geovisualization* 10(2):112–122

Dykes, J., Purves, R., Edwardes, A. and Wood, J. 2008. "Exploring volunteered geographic information to describe place: visualization of the 'Geograph British Isles' collection." In *Proceedings of the GIS Research UK 16th Annual Conference GISRUK*. pp. 256–267

Elotheos, S. 2013. "Photo count but no result - Google Groups."
https://groups.google.com/forum/\#!topic/panoramio-api/wjD9p6jOOE8

Eppstein, D. 1998. "Finding the k shortest paths." *SIAM Journal on computing* 28(2):652–673
http://epubs.siam.org/doi/abs/10.1137/S0097539795290477

EPSG Registry 2014. "EPSG:3857."
http://bit.ly/EPSG_3857

EuroStat 2014. "Population and social conditions."
http://epp.eurostat.ec.europa.eu/portal/page/portal/population/data

Few, S. 2007. "Save the pies for dessert."
http://bit.ly/1LRV9rP

Finkel, R. and Bentley, J. 1974. "Quad trees a data structure for retrieval on composite keys." *Acta Informatica* 4(1):1–9
http://link.springer.com/10.1007/BF00288933

Fischer, E. 2010. "Locals and Tourists – an album on Flickr."
https://www.flickr.com/photos/walkingsf/sets/72157624209158632/

Flickr 2007. "The Help Forum: Flickr never returns more than 4,000 results on any search."
https://www.flickr.com/help/forum/62077/

Forbes 2014. "Most Visited Cities In The World 2014."
http://www.forbes.com/pictures/efik45fjjik/no-1-london/

Friedman, N., Geiger, D. and Goldszmidt, M. 1997. "Bayesian network classifiers." *Machine learning* 29(2-3):131–163

Fry, B. 2008. *Visualizing Data: Exploring and Explaining Data with the Processing Environment*. Beijing; Cambridge: O'Reilly Media, Inc.

Fulgham, B. and Gouy, I. 2014. "Computer Language Benchmarks Game."
http://benchmarksgame.alioth.debian.org/

Gao, M., Hua, X. and Jain, R. 2011. "WonderWhat: real-time event determination from photos." In *Proceedings of the 20th international conference companion on World wide web*. pp. 37–38

Geograph 2014*a*. "Data Dumps."
http://data.geograph.org.uk/dumps/columns.html

Geograph 2014*b*. "How are pictures moderated?"
http://www.geograph.org.uk/article/Geograph-or-supplemental

Gillan, A. 2009. "Heavy snow due to continue all week." *The Guardian*
http://www.theguardian.com/uk/2009/feb/02/snow-london-travel-chaos

Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. and Roberts, J. 2011. "Visual comparison for information visualization." *Information Visualization* 10(4):289–309
http://ivi.sagepub.com/lookup/doi/10.1177/1473871611416549

Goodwin, S., Dykes, J., Jones, S., Dillingham, I., Dove, G., Duffy, A., Kachkaev, A., Slingsby, A. and Wood, J. 2013. "Creative User-Centered Visualization Design for Energy Analysts and Modelers." *IEEE Transactions on Visualization and Computer Graphics* 19(12):2516–2525
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6634166

Google 2013. "Picasa Web Albums and Google+ - Picasa and Picasa Web Albums Help."
https://support.google.com/picasa/answer/1321133

Greater London Authority and London Development Agency 2003. *Valuing greenness: green spaces, house prices and Londoners' priorities.* London: Greater London Authority.

Grgic, M. and Delac, K. 2007. "Face Recognition Homepage - Algorithms."
http://www.face-rec.org/algorithms/

Gruber, J. and Prodanovic, M. 2012. "Residential Energy Load Profile Generation Using a Probabilistic Approach.". IEEE pp. 317–322
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6410171

Haklay, M. 2010. "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets." *Environment and planning. B, Planning & design* 37(4):682

Harley, J. 1976. *Ordnance Survey Maps: A Descriptive Manual.* [S.l.]: Ordnance Survey.

Hart, P., Nilsson, N. and Raphael, B. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4082128

Hauff, C. 2013. "A study on the accuracy of Flickr's geotag data." In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM pp. 1037–1040
http://dl.acm.org/citation.cfm?id=2484154

Hawick, A., Coddington, P. and James, H. 2003. "Distributed frameworks and parallel algorithms for processing large-scale geographic data." *Parallel Computing* 29(10):1297–1333
http://linkinghub.elsevier.com/retrieve/pii/S0167819103001054

Heer, J. and Robertson, G. 2007. "Animated transitions in statistical data graphics." *Visualization and Computer Graphics, IEEE Transactions on* 13(6):1240–1247
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4376146

Herbst, J., McGrath, S. and Borak, J. 2008. "Method of operating a navigation system using images."
http://www.google.com/patents/US7460953

Hibernate 2014. "What is Object/Relational Mapping? – Hibernate ORM."
http://hibernate.org/orm/what-is-an-orm/

Hietanen, H., Athukorala, K. and Salovaara, A. 2011. "What's with the free images? A study of Flickr's creative commons attribution images." *Proc. MindTrek 2011*

Hile, H., Grzeszczuk, R., Liu, A., Vedantham, R., Košecka, J. and Borriello, G. 2009. "Landmark-based pedestrian navigation with enhanced spatial reasoning." *Pervasive Computing* pp. 59–76
http://www.springerlink.com/index/76R21777K4WX6TJ0.pdf

Hile, H., Vedantham, R., Cuellar, G., Liu, A., Gelfand, N., Grzeszczuk, R. and Borriello, G. 2008. "Landmark-based pedestrian navigation from collections of geotagged photos." In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*. pp. 145–152
`http://dl.acm.org/citation.cfm?id=1543167`

Hochmair, H. 2010. "Spatial association of geotagged photos with scenic locations." In *Proceedings of the Geoinformatics Forum Salzburg, A. Car, G. Griesebner and J. Strobl (Eds.)*. pp. 91–100
`http://www.agit.at/php_files/myagit/papers/2010/8134.pdf`

Hochmair, H. and Navratil, G. 2008. "Computation of Scenic Routes in Street Networks." *Geospatial Crossroads@ GI_Forum* 8:124–133

Hochmair, H. and Zielstra, D. 2011. "Digital Street Data: Free versus Proprietary." *GIM International* 25(7):29–33

Hochmair, H. and Zielstra, D. 2012. "Positional accuracy of Flickr and Panoramio images in Europe." In *Geospatial Crossroads@ GI Forum'12: Proceedings of the Geoinformatics Forum. Heidelberg, Germany, Wichman*. pp. 14–23
`http://bit.ly/1KyX0EQ`

Inselberg, A. and Dimsdale, B. 1990. "Parallel coordinates: a tool for visualizing multi-dimensional geometry." In *Proceedings of the 1st conference on Visualization '90*. VIS '90 Los Alamitos, CA, USA: IEEE Computer Society Press pp. 361–378
`http://dl.acm.org/citation.cfm?id=949531.949588`

ITU-T 1994. "X.200 (1994) ISO/IEC 7498-1: 1994." *Information technology–Open Systems Interconnection–Basic Reference Model: The basic model*

Jacobson, R. 2000. *The Manual of Photography: Photographic and Digital Imaging*. Media Manual Focal Press.
`http://books.google.co.uk/books?id=5YnWKdBWEd8C`

Jain, R. and Sinha, P. 2008. *Photo classification using optical parameters of camera from exif metadata*. Google Patents. US Patent App. 12/110,065.
http://www.google.ca/patents/US20080292196

Jain, S., Seufert, S. and Bedathur, S. 2010. "Antourage: mining distance-constrained trips from flickr." In *Proceedings of the 19th international conference on World wide web*. ACM Press pp. 1121–1122
http://portal.acm.org/citation.cfm?doid=1772690.1772834

Kachkaev, A., Wood, J. and Dykes, J. 2014. "Glyphs for Exploring Crowd-sourced Subjective Survey Classification." *Computer Graphics Forum* 33(3):311–320
http://onlinelibrary.wiley.com/doi/10.1111/cgf.12387/abstract

Kantor, M. and Rosenwein, M. 1992. "The orienteering problem with time windows." *Journal of the Operational Research Society* pp. 629–635
http://www.jstor.org/stable/10.2307/2583018

Keim, D., Andrienko, G., Fekete, J., Görg, C., Kohlhammer, J. and Melançon, G. 2008. *Visual analytics: Definition, process, and challenges*. Springer.
http://link.springer.com/chapter/10.1007/978-3-540-70956-5_7

Kennedy, L. and Naaman, M. 2008. "Generating diverse and representative image search results for landmarks." In *Proceedings of the 17th international conference on World Wide Web*. ACM pp. 297–306
http://dl.acm.org/citation.cfm?id=1367539

Kisilevich, S., Mansmann, F., Bak, P., Keim, D. and Tchaikin, A. 2010. "Where Would You Go on Your Next Vacation? A Framework for Visual Exploration of Attractive Places.". IEEE pp. 21–26
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5437985

Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R. 2009. "Controlled experiments on the web: survey and practical guide." *Data Mining and Knowledge Discovery* 18(1):140–181
http://link.springer.com/10.1007/s10618-008-0114-1

Konica Minolta 2013. "Luminance meter LS-100, LS-110 (instruction manual)." `http://bit.ly/1NbSDOW`

Kurashima, T., Iwata, T., Irie, G. and Fujimura, K. 2010. "Travel route recommendation using geotags in photo sharing sites." In *Proceedings of the 19th ACM international conference on Information and knowledge management*. pp. 579–588

Kurashima, T., Iwata, T., Irie, G. and Fujimura, K. 2012. "Travel route recommendation using geotagged photos." *Knowledge and Information Systems* `http://www.springerlink.com/index/10.1007/s10115-012-0580-z`

L. S. R. Online 2010. "Place Survey." `http://www.lsr-online.org/placesurvey.html`

Lee, B., Chen, W. and Chang, E. 2006. "A scalable service for photo annotation, sharing, and search." In *Proceedings of the 14th annual ACM international conference on Multimedia*. ACM pp. 699–702 `http://dl.acm.org/citation.cfm?id=1180787`

Lee, C., Greene, D. and Cunningham, P. 2011. "Detecting Grand Tours of Europe with Geo-Tags."

Lerner, J., D. Wagner, K. Zweig, D. Hutchison, T. Kanade, J. Kittler, J. Kleinberg, F. Mattern, J. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Vardi and G. Weikum, eds 2009. *Algorithmics of Large and Complex Networks*. Vol. 5515 of *Lecture Notes in Computer Science* Berlin, Heidelberg: Springer Berlin Heidelberg. `http://www.springerlink.com/index/10.1007/978-3-642-02094-0`

Lie, A., Kehrer, J. and Hauser, H. 2009. "Critical design and realization aspects of glyph-based 3D data visualization." In *Proceedings of the 25th Spring Conference on Computer Graphics*. ACM pp. 19–26

Lind, J. 2000. "Photographic Science: Exposure." `http://johnlind.tripod.com/science/scienceexposure.html`

Liu, Y. and Li, X. 2013. "Indoor-outdoor image classification using mid-level cues." In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE pp. 1–5
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6694294

Lu, X., Wang, C., Yang, J., Pang, Y. and Zhang, L. 2010. "Photo2Trip: generating travel routes from geo-tagged photos for trip planning.". ACM Press p. 143
http://portal.acm.org/citation.cfm?doid=1873951.1873972

Maguire, E., Rocca-Serra, P., Sansone, S., Davies, J. and Chen, M. 2012. "Taxonomy-Based Glyph Design—with a Case Study on Visualizing Workflows of Biological Experiments." *Visualization and Computer Graphics, IEEE Transactions on* 18(12):2603–2612

Mamei, M., Rosi, A. and Zambonelli, F. 2010. "Automatic analysis of geotagged photos for intelligent tourist services." In *Intelligent Environments (IE), 2010 Sixth International Conference on*. IEEE pp. 146–151
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5673673

Martin, J. 1983. *Managing the data-base environment*. Englewood Cliffs, N.J: Prentice-Hall.

Mashape 2013. "List of 50+ Face Detection / Recognition APIs, libraries, and software."
http://blog.mashape.com/list-of-50-face-detection-recognition-apis/

McHugh, J. 1990. *Algorithmic Graph Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Memon, I., Chen, L., Majid, A., Lv, M., Hussain, I. and Chen, G. 2014. "Travel Recommendation Using Geo-tagged Photos in Social Media for Tourist." *Wireless Personal Communications*
http://link.springer.com/10.1007/s11277-014-2082-7

Met Office 2014. "London Climate."
http://www.metoffice.gov.uk/public/weather/climate/gcpvj0v07

Microsoft 2010. "Quick start: Apply conditional formatting."
http://bit.ly/1i5Y7PX

Migliorini, S., Gambini, M., Belussi, A., Negri, M. and Pelagatti, G. 2011. "Workflow technology for geo-processing: the missing link." In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*. ACM p. 36 `http://dl.acm.org/citation.cfm?id=1999356`

Movable Type Scripts 2012. "Calculate distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript." `http://www.movable-type.co.uk/scripts/latlong.html`

MVA Consultency 2010. Shared Space: Operational Assessment. Technical report. `http://assets.dft.gov.uk/publications/ltn-01-11/ltn-1-11-quantitative.pdf`

mySociety 2009. "Scenic or not?" `http://www.dxw.com/portfolio/mysociety-scenic-or-not/`

NGA 2004. "World Geodetic System 1984." `http://earth-info.nga.mil/GandG/wgs84/index.html`

Nielsen, J. 2000. "Why You Only Need to Test with 5 Users." `http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`

Nielsen, J. 2006. "The 90-9-1 Rule for Participation Inequality in Social Media and Online Communities." *Alertbox* (9 October)

O'Brien, O. 2013. "This Place, Visualisation of 2011 Census data for England and Wales." `http://casa.oobrien.com/thisplace/`

Office for National Statistics 2011. "2011 Census Interactive." `http://bit.ly/LPptce`

Okuyama, K. and Yanai, K. 2010. "A Travel Planning System Based on Travel Trajectories Extracted from a Large Number of Geotagged Photos on the Web." `http://img.cs.uec.ac.jp/pub/conf11/111220yanai_2.pdf`

ONS 2014. Monthly Overseas Travel and Tourism in London. Technical report. `http://bit.ly/1QiwOwX`

OpenCV 2014. "Face Detection using Haar Cascades."
http://bit.ly/1EKt3PA

OpenStreetMap 2012. "Copyright and License."
http://www.openstreetmap.org/copyright

Oracle 2014. "MySQL 5.6 Reference Manual."
http://dev.mysql.com/doc/refman/5.6/en/index.html

Ordnance Survey 2012. "National Grid used on Ordnance Survey maps - the definitive guide."
http://bit.ly/os_national_grid

OSM-Wiki 2014*a*. "Key:highway."
http://wiki.openstreetmap.org/wiki/Highway

OSM-Wiki 2014*b*. "Routing/online routers."
http://wiki.openstreetmap.org/wiki/Routing/OnlineRouters

Pahlavani, P., Samadzadegan, F. and Delavar, M. 2006. "A GIS-Based Approach for Urban Multi-criteria Quasi Optimized Route Guidance by Considering Unspecified Site Satisfaction." In *Geographic Information Science*, ed. D. Hutchison, T. Kanade, J. Kittler, J. Kleinberg, F. Mattern, J. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Vardi, G. Weikum, M. Raubal, H. Miller, A. Frank and M. Goodchild. Vol. 4197 Berlin, Heidelberg: Springer Berlin Heidelberg pp. 287–303.
http://www.springerlink.com/index/10.1007/11863939_19

Panoramio 2014. "Photo acceptance policy."
http://www.panoramio.com/help/acceptance_policy/

Peca, I., Zhi, H., Vrotsou, K., Andrienko, N. and Andrienko, G. 2011. "KD-photomap: Exploring photographs in space and time." In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. pp. 291–292

Pillai, I., Satta, R., Fumera, G. and Roli, F. 2011. "Exploiting depth information for indoor-outdoor scene classification." In *Image Analysis and Processing–ICIAP 2011*. Springer pp. 130–139.

Popescu, A. and Grefenstette, G. 2011. "Mining social media to create personalized recommendations for tourist visits." In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*. p. 37

PostgreSQL 2014. "Manuals."
`http://www.postgresql.org/docs/manuals/`

Purves, R., Edwardes, A. and Wood, J. 2011. "Describing place through user generated content." *First Monday* 16(9)
`http://firstmonday.org/ojs/index.php/fm/article/view/3710/3035`

Quack, T., Leibe, B. and Van Gool, L. 2008. "World-scale mining of objects and events from community photo collections." In *Proceedings of the 2008 international conference on Content-based image and video retrieval*. pp. 47–56

Quercia, D., Schifanella, R. and Aiello, L. 2014. "The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city.". ACM Press pp. 116–125
`http://dl.acm.org/citation.cfm?doid=2631775.2631799`

Ramblers' Association 2010. Walking facts and figures 2: Participation in walking. Technical report.

Ramblers' Association 2012. "Walking for Health - Health walks, walking clubs, walking exercise."
`http://www.walkingforhealth.org.uk/`

Ramm, F., Topf, J. and Chilton, S. 2011. *OpenStreetMap: Using and Enhancing the Free Map of the World*. UIT Cambridge.
`http://books.google.co.uk/books?id=AnCNQQAACAAJ`

Rao, R. and Card, S. 1994. "The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information." In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM pp. 318–322

Raskin, J. 2000. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press Series Addison-Wesley.
http://books.google.co.uk/books?id=D39vjmLfO3kC

Rattenbury, T., Good, N. and Naaman, M. 2007. "Towards automatic extraction of event and place semantics from flickr tags." In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 103–110

Riehle, D. 2000. Framework design PhD thesis, Diss. Technische Wissenschaften ETH Zürich, Nr. 13509, 2000.
http://e-collection.library.ethz.ch/view/eth:23315

Rivest, R. 1992. "The MD5 message-digest algorithm."

Rose, J. and Wong, P. 2000. "DriftWeed: a visual metaphor for interactive analysis of multivariate data.". Vol. 3960 pp. 114–121
http://dx.doi.org/10.1117/12.378887

Salkind, N. and Rasmussen, K. 2007. *Encyclopedia of measurement and statistics*. Thousand Oaks, Calif.: SAGE Publications.
http://www.credoreference.com/book/sagemeasure

Samet, H. 1990*a*. *Applications of spatial data structures : computer graphics, image processing, and GIS*. Reading, Mass.: Addison-Wesley.

Samet, H. 1990*b*. *The design and analysis of spatial data structures*. Reading, Mass.: Addison-Wesley.

Sapsford, R. 2007. *Survey research*. 2nd ed ed. London ; Thousand Oaks, Calif: Sage Publications.

Scott, M. 2009. *Programming Language Pragmatics*. Elsevier Science. `http://books.google.co.uk/books?id=GBISkhhrHh8C`

Serrano, N., Savakis, A. and Luo, J. 2002. "A computationally efficient approach to indoor/outdoor scene classification." In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 4 IEEE pp. 146–149

Sizo, R. 2013. "Panoramio API download limit - Google Groups." `http://bit.ly/1NjMi5v`

SQLite 2014. "Documentation." `http://www.sqlite.org/docs.html`

Stringhini, G., Egele, M., Kruegel, C. and Vigna, G. 2012. "Poultry markets: on the underground economy of twitter followers.". ACM Press p. 1 `http://dl.acm.org/citation.cfm?doid=2342549.2342551`

Strous, L. 2014. "Astronomy Answers: Position of the Sun." `http://aa.quae.nl/en/reken/zonpositie.html`

Sugiyama, T., Leslie, E., Giles-Corti, B. and Owen, N. 2008. "Associations of neighbourhood greenness with physical and mental health: do walking, social coherence and local social interaction explain the relationships?" *Journal of Epidemiology and Community Health* 62(5):e9–e9

SurveyMonkey 2014. "How can I revise the design of my survey to improve response rates?" `http://svy.mk/1JK8MGR`

Szummer, M. and Picard, R. 1998. "Indoor-outdoor image classification." In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*. pp. 42–51

Tastle, W. and Wierman, M. 2006. "An information theoretic measure for the evaluation of ordinal scale data." *Behaviour Research Methods* 38(3):487–494

Tchaikin, A. 2008. “Building photo density maps with Panoramio - ProjeKCts.”
http://bit.ly/1JMB71s

The Open Group 2010. “What Is SOA?”
http://www.opengroup.org/soa/source-book/soa/soa.htm\#soa_definition

Thomas, J. and Cook, K. 2006. “A visual analytics agenda.” *IEEE Computer Graphics and Applications* 26(1):10–13
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1573625

Transport for London 2012. “Legible London.”
http://tfl.gov.uk/microsites/legible-london/

Transport for London 2014. “Congestion Charge zone.”
http://tfl.gov.uk/modes/driving/congestion-charge/congestion-charge-zone

Trautman, P. and Mostek, J. 2000. “Scalability and Performance in Modern Filesystems.” *SGI white paper*
http://linux-xfs.sgi.com/projects/xfs/papers/xfs_white/xfs_white_paper.html

Tufte, E. 1983. *The visual display of quantitative information*. Number v. 914 *in* “The Visual Display of Quantitative Information” Graphics Press.
http://books.google.co.uk/books?id=tWpHAAAAMAAJ

United States Census Bureau 2013. “Data Visualization Gallery.”
http://www.census.gov/dataviz/

U.S. Air Force 2014. “GPS.gov: GPS Accuracy.”
http://www.gps.gov/systems/gps/performance/accuracy/

Viola, P. and Jones, M. 2001. “Rapid object detection using a boosted cascade of simple features.” In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1 IEEE pp. I–511
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=990517

Walk4Life 2012. "Welcome to Walk4Life."
`http://www.walk4life.info/`

WalkEngland 2012. "Welcome to Walk England."
`http://www.walkengland.org.uk/`

Walker, R., Slingsby, A., Dykes, J., Xu, K., Wood, J., Nguyen, P., Stephens, D., Wong, B. and Zheng, Y. 2013. "An extensible framework for provenance in human terrain visual analytics." *Visualization and Computer Graphics, IEEE Transactions on* 19(12):2139–2148

WalkLondon 2012. "Welcome to Walk London."
`http://www.walklondon.org.uk/`

Ward, M. 2002. "A taxonomy of glyph placement strategies for multidimensional data visualization." *Information Visualization* 1(3-4):194–210

Ward, M. 2008. "Multivariate data glyphs: Principles and practice." In *Handbook of data visualization*. Springer pp. 179–198.

Watkins, J. 2014. "pthreads - Share Nothing, Do Everything."
`http://pthreads.org/`

Watt, D. 2006. *Programming Language Design Concepts*. Wiley.
`http://books.google.co.uk/books?id=vogP3P2L4tgC`

Wickham, H., Hofmann, H., Wickham, C. and Cook, D. 2012. "Glyph-maps for visually exploring temporal patterns in climate data and models." *Environmetrics* 23(5):382–393
`http://doi.wiley.com/10.1002/env.2152`

Wilkinson, L. and Wills, G. 2005. *The Grammar of Graphics*. Statistics and Computing Springer.
`http://books.google.co.uk/books?id=_kRX4LoFfGQC`

Wong, P. and Thomas, J. 2004. "Visual Analytics." *IEEE Computer Graphics and Applications* 24(5):20–21
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1333623

Woodhouse 2012. "Legible Brighton."
http://www.woodhouse.co.uk/brighton.html

Yamasaki, T., Gallagher, A. and Chen, T. 2013. "Geotag-based travel route recommendation featuring seasonal and temporal popularity." In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*. IEEE pp. 1–4
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6782963

Yin, H., Wang, C., Yu, N. and Zhang, L. 2012. "Trip Mining and Recommendation from Geo-tagged Photos.". IEEE pp. 540–545
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6266441

Zeng, W. and Church, R. 2009. "Finding shortest paths on real road networks: the case for A*." *International Journal of Geographical Information Science* 23(4):531–543
http://www.tandfonline.com/doi/abs/10.1080/13658810801949850

Zheng, Y., Yan, S., Zha, Z., Li, Y., Zhou, X., Chua, T. and Jain, R. 2013. "GPSView: A Scenic Driving Route Planner." *ACM Trans. Multimedia Comput. Commun. Appl.* 9(1):3:1–3:18
http://doi.acm.org/10.1145/2422956.2422959

Zielstra, D. and Hochmair, H. 2013. "Positional accuracy analysis of Flickr and Panoramio images for selected world regions." *Journal of Spatial Science* 58(2):251–273
http://www.tandfonline.com/doi/abs/10.1080/14498596.2013.801331

Appendix A

List of implemented software

Dataset Abstraction Framework (DAF) php postgresql twig

A framework for processing multiple arbitrary datasets from different domains. Works on top of Symfony 2. See Section 3.2 on page 60 for details.

`https://github.com/kachkaev/KachkaevDatasetAbstractionBundle`

Photo-based routing research package java php postgresql postgis pgrouting r twig

Based on the Dataset Abstraction Framework, this software comprises all application-specific code that was written to conduct the experiments described in Chapters 4 and 5. Some components such as quad-based data processor (page 83) may be reused for a wider range of tasks.

Photo content assessment tool (survey glyphs) html css javascript php mysql

Survey data analysis interface as introduced in Section 4.4 on page 176, questionnaire for http://www.photoassessment.org/, collected responses.

`https://github.com/kachkaev/survey-glyphs`

PHP-R php r

A library that provides ability to execute R scripts from PHP 5.3+. Available as a Composer package and as a Symfony 2 bundle. See page 86 for details.

`https://github.com/kachkaev/php-r`

`https://github.com/kachkaev/KachkaevPHPRBundle`

CICommandLineFaceDetector objective-c

A simple command line tool to detect faces in photographs using Apple's Core Image library. Mentioned in Subsection 4.5.5 on page 205.

`https://github.com/kachkaev/CICommandLineFaceDetector`

Appendix B

Commands available in the Dataset Abstraction Framework

```
$ cd path/to/a/daf-based-project
```

```
# List of all available commands in the Dataset Abstraction Framework Core
$ app/console list daf

Available commands for the "daf" namespace:
  daf:datasets:backup                           Dumps selected dataset into a backup file
  daf:datasets:components:attributes:copy       Copies given attributes from the same
      component of another dataset
  daf:datasets:components:attributes:delete     Deletes dataset component attribute
  daf:datasets:components:attributes:init       Initialises one or several similar
      attributes in the component
  daf:datasets:components:attributes:list       Lists attributes in the dataset component
  daf:datasets:components:attributes:rename     Renames the dataset component attribute
  daf:datasets:components:attributes:reset      Resets an attribute of the given dataset
      component
  daf:datasets:components:attributes:update     Updates given attributes of the given
      dataset component
  daf:datasets:components:delete                Deletes dataset component
  daf:datasets:components:init                  Initialises dataset component
  daf:datasets:components:list                  Lists existing components in the dataset
  daf:datasets:components:records:clean         Removes records from the component (all
      or a filtered subset)
  daf:datasets:components:records:copy          Copies records into the component from
      another dataset
  daf:datasets:components:records:count         Counts records in the component (all or
      a filtered subset)
  daf:datasets:components:records:populate      Populates the given component with
      records using a corresponding service
  daf:datasets:components:reset                 Deletes all data in the dataset
      component and recreates it
  daf:datasets:delete                           Deletes the given dataset
  daf:datasets:duplicate                        Renames the given dataset
  daf:datasets:init                             Initialises an empty dataset
  daf:datasets:list                             Lists existing datasets in the given
      domain
  daf:datasets:properties:copy                  Copies properties from the origin
      dataset to the given dataset
  daf:datasets:properties:list                   Lists existing dataset properties
  daf:datasets:properties:set                   Sets a single dataset property
  daf:datasets:rename                           Renames the given dataset
  daf:datasets:restore                          Restores a dataset from a given dump file
  daf:db:dump-sql                               Runs the query from template and saves
      the result into a file
  daf:db:init                                   Initialises the project's database
  daf:db:query-to-file                          Saves the result of a query to a file
  daf:domains:delete                            Deletes the given data domain in the
      project's database
  daf:domains:init                              Initialises given data domain in the
      project's database
  daf:domains:list                              Lists existing domains
  daf:domains:update-functions                  Updates functions in a given domain
  daf:domains:update-types                      Updates types in a given domain
```

```
# Details of the command that initialises a new dataset
$ app/console daf:datasets:init --help

Usage:
 daf:datasets:init dataset-full-name [dataset-type]

Arguments:
 dataset-full-name     Full name of the dataset to work with (i.e.
     domain_name.dataset_name)
 dataset-type          Type of the new dataset

Options:
 --help (-h)           Display this help message.
 --quiet (-q)          Do not output any message.
 --verbose (-v|vv|vvv) Increase the verbosity of messages: 1 for normal output, 2 for
     more verbose output and 3 for debug
 --version (-V)        Display this application version.
 --ansi                Force ANSI output.
 --no-ansi             Disable ANSI output.
 --no-interaction (-n) Do not ask any interactive question.
 --shell (-s)          Launch the shell.
 --process-isolation   Launch commands from shell as a separate process.
 --env (-e)            The Environment name. (default: "dev")
 --no-debug            Switches off debug mode.
```

```
# Details of the command that allows new component attributes to be created
$ app/console daf:datasets:components:attributes:init --help

Usage:
 daf:datasets:components:attributes:init dataset-full-name component-name
     attribute-names attribute-definition [attribute-comment]

Arguments:
 dataset-full-name     Full name of the dataset to work with (i.e.
     domain_name.dataset_name)
 component-name        Name of the component
 attribute-names       Comma-separated names of attributes to create
 attribute-definition  Definition of all created attributes (postgres colum types)
 attribute-comment     Attribute comment (to be saved in postgres db)

Options:
 --help (-h)           Display this help message.
 ...
```

```
# Details of the command for updating (deriving) attributes in a component
$ app/console daf:datasets:components:attributes:update --help

Usage:
 daf:datasets:components:attributes:update [--ids="..."] [--filter="..."]
     [--chunk-size="..."] dataset-full-name component-name attribute-names

Arguments:
 dataset-full-name     Full name of the dataset to work with (i.e.
     domain_name.dataset_name)
 component-name        Name of the component
 attribute-names       Comma-separated names of attributes to update

Options:
 --ids                 ids of records to update attributes for (alternative to
     --filter)
 --filter              sql WHERE to filter records and get their ids (alternative to
     --ids)
 --chunk-size          number of records in a batch
 --help (-h)           Display this help message.
 ...
```

Appendix C

Survey results analysis

	luminance threshold (step = 0.25)																			
	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6									
	natural counts																			
all photos																				
positives (P)	343	342	339	338	337	336	336	332	332	330	326	324	320	316	311	300	288	279	262	247	231
negatives (N)	138	141	144	152	161	166	169	174	177	177	179	184	189	191	194	194	195	198	201	203	203
false positives (FP)	85	82	79	71	62	57	54	49	46	46	44	39	34	32	29	29	28	25	22	20	20
false negatives (FN)	19	20	23	24	25	26	26	30	30	32	36	38	42	46	51	62	74	83	100	115	131
distance to ideal	87	84	82	75	67	63	60	57	55	56	57	54	54	56	59	68	79	87	102	117	133
chi square	225	230	229	248	271	283	292	296	305	299	294	304	309	305	301	274	249	240	216	195	169
FP+FN	104	102	102	95	87	83	80	79	76	78	80	77	76	78	80	91	102	108	122	135	151
Flickr																				
P	129	129	128	128	127	126	126	125	125	124	124	123	121	118	116	110	104	101	93	83	77
N	101	104	106	112	119	123	125	128	128	128	128	131	135	135	137	137	137	138	140	142	142
FP	47	44	42	36	29	25	23	20	20	20	20	17	13	13	11	11	11	10	8	6	6
FN	13	13	14	14	15	16	16	17	17	18	18	19	21	24	26	32	38	41	49	59	65
distance to ideal	49	46	44	39	33	30	28	26	26	27	27	25	25	27	28	34	40	42	50	59	65
chi square	106	112	114	128	142	150	155	161	161	158	158	164	170	162	162	146	131	127	115	101	89
FP+FN	60	57	56	50	44	41	39	37	37	38	38	36	34	37	37	43	49	51	57	65	71
Panoramio																				
P	214	213	211	210	210	210	210	207	207	206	202	201	199	198	195	190	184	178	169	164	154
N	37	37	38	40	42	43	44	46	49	49	51	53	54	56	57	57	58	60	61	61	61
FP	38	38	37	35	33	32	31	29	26	26	24	22	21	19	18	18	17	15	14	14	14
FN	6	7	9	10	10	10	10	13	13	14	18	19	21	22	25	30	36	42	51	56	66
distance to ideal	38	39	38	36	34	34	33	32	29	30	30	29	30	29	31	35	40	45	53	58	67
chi square	98	94	91	95	102	106	110	107	119	116	112	117	115	120	116	105	96	92	80	73	60
FP+FN	44	45	46	45	43	42	41	42	39	40	42	41	42	41	43	48	53	57	65	70	80
	percentage																			
all photos																				
P	58.6	58.5	57.9	57.8	57.6	57.4	57.4	56.8	56.8	56.4	55.7	55.4	54.7	54.0	53.2	51.3	49.2	47.7	44.8	42.2	39.5
N	23.6	24.1	24.6	26.0	27.5	28.4	28.9	29.7	30.3	30.3	30.6	31.5	32.3	32.6	33.2	33.2	33.3	33.8	34.4	34.7	34.7
FP	14.5	14.0	13.5	12.1	10.6	9.7	9.2	8.4	7.9	7.9	7.5	6.7	5.8	5.5	5.0	5.0	4.8	4.3	3.8	3.4	3.4
FN	3.2	3.4	3.9	4.1	4.3	4.4	4.4	5.1	5.1	5.5	6.2	6.5	7.2	7.9	8.7	10.6	12.6	14.2	17.1	19.7	22.4
distance to ideal	14.9	14.4	14.1	12.8	11.4	10.7	10.2	9.8	9.4	9.6	9.7	9.3	9.2	9.6	10.0	11.7	13.5	14.8	17.5	20.0	22.7
FP+FN	17.8	17.4	17.4	16.2	14.9	14.2	13.7	13.5	13.0	13.3	13.7	13.2	13.0	13.3	13.7	15.6	17.4	18.5	20.9	23.1	25.8
Flickr																				
P	44.5	44.5	44.1	44.1	43.8	43.4	43.4	43.1	43.1	42.8	42.8	42.4	41.7	40.7	40.0	37.9	35.9	34.8	32.1	28.6	26.6
N	34.8	35.9	36.6	38.6	41.0	42.4	43.1	44.1	44.1	44.1	44.1	45.2	46.6	46.6	47.2	47.2	47.2	47.6	48.3	49.0	49.0
FP	16.2	15.2	14.5	12.4	10.0	8.6	7.9	6.9	6.9	6.9	6.9	5.9	4.5	4.5	3.8	3.8	3.8	3.4	2.8	2.1	2.1
FN	4.5	4.5	4.8	4.8	5.2	5.5	5.5	5.9	5.9	6.2	6.2	6.6	7.2	8.3	9.0	11.0	13.1	14.1	16.9	20.3	22.4
distance to ideal	16.8	15.8	15.3	13.3	11.3	10.2	9.7	9.1	9.1	9.3	9.3	8.8	8.5	9.4	9.7	11.7	13.6	14.6	17.1	20.4	22.5
FP+FN	20.7	19.7	19.3	17.2	15.2	14.1	13.4	12.8	12.8	13.1	13.1	12.4	11.7	12.8	12.8	14.8	16.9	17.6	19.7	22.4	24.5
Panoramio																				
P	72.5	72.2	71.5	71.2	71.2	71.2	71.2	70.2	70.2	69.8	68.5	68.1	67.5	67.1	66.1	64.4	62.4	60.3	57.3	55.6	52.2
N	12.5	12.5	12.9	13.6	14.2	14.6	14.9	15.6	16.6	16.6	17.3	18.0	18.3	19.0	19.3	19.3	19.7	20.3	20.7	20.7	20.7
FP	12.9	12.9	12.5	11.9	11.2	10.8	10.5	9.8	8.8	8.8	8.1	7.5	7.1	6.4	6.1	6.1	5.8	5.1	4.7	4.7	4.7
FN	2.0	2.4	3.1	3.4	3.4	3.4	3.4	4.4	4.4	4.7	6.1	6.4	7.1	7.5	8.5	10.2	12.2	14.2	17.3	19.0	22.4
distance to ideal	13.0	13.1	12.9	12.3	11.7	11.4	11.0	10.8	9.9	10.0	10.2	9.9	10.1	9.9	10.4	11.9	13.5	15.1	17.9	19.6	22.9
FP+FN	14.9	15.3	15.6	15.3	14.6	14.2	13.9	14.2	13.2	13.6	14.2	13.9	14.2	13.9	14.6	16.3	18.0	19.3	22.0	23.7	27.1

Table C.1: Comparison of different values for photo luminance threshold.

		all photos								Flickr								Geograph								Panoramio							
photo size	min face size	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh
presence of faces																																	
P																																	
240	0	124	54	44	52	30	30	43	44	106	49	43	50	28	29	42	43	4	0	0	0	0	0	0	0	14	5	1	2	2	1	1	1
240	5	91	48	44	50	29	29	43	44	88	47	43	49	28	29	42	43	0	0	0	0	0	0	0	0	3	1	1	1	1	0	1	1
240	10	52	33	32	34	19	19	31	32	51	33	32	34	19	19	31	32	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
240	20	11	7	5	5	2	5	7	7	11	7	5	5	2	5	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
240	30	6	1	2	2	1	2	2	2	6	1	2	2	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
500	0	124	95	78	83	55	59	63	65	106	83	70	72	50	55	61	62	4	3	3	4	2	2	0	1	14	9	5	7	3	2	2	2
500	5	91	77	63	66	46	49	59	61	88	74	62	65	45	49	58	60	0	0	0	0	0	0	0	0	3	3	1	1	1	0	1	1
500	10	52	38	37	37	25	27	34	35	51	38	37	37	25	27	34	35	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
500	20	11	5	7	6	2	4	8	8	11	5	7	6	2	4	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
500	30	6	1	2	2	1	2	4	3	6	1	2	2	1	2	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1024	0	124	119	102	117	77	87	62	67	106	103	88	100	66	77	61	64	4	3	3	4	2	2	0	1	14	13	11	13	9	8	1	2
1024	5	91	86	68	80	50	60	58	61	88	84	67	78	49	59	57	60	0	0	0	0	0	0	0	0	3	2	1	2	1	1	1	1
1024	10	52	41	34	39	24	26	34	36	51	41	34	39	24	25	34	36	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
1024	20	11	5	6	7	3	5	6	7	11	5	6	7	3	5	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1024	30	6	1	2	1	1	2	3	3	6	1	2	1	1	2	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N																																	
240	0	765	693	734	717	759	736	763	764	186	159	171	165	185	174	184	185	296	279	290	284	295	290	296	296	283	255	273	268	279	272	283	283
240	5	798	720	767	748	791	768	796	797	204	175	189	182	203	192	202	203	300	283	294	288	299	294	300	300	294	262	284	278	289	282	294	294
240	10	837	745	799	780	826	799	829	830	241	198	219	212	236	220	233	234	300	284	294	288	299	295	300	300	296	263	286	280	291	284	296	296
240	20	878	829	862	846	872	860	876	877	281	255	273	263	279	268	279	280	300	291	296	294	299	298	300	300	297	283	293	289	294	294	297	297
240	30	883	865	876	869	879	879	882	881	286	275	283	277	285	282	285	284	300	299	299	298	300	300	300	300	297	291	294	294	294	297	297	297
500	0	765	326	559	457	714	594	759	750	186	95	149	128	175	139	185	182	296	95	186	137	269	219	294	292	283	136	224	192	270	236	280	276
500	5	798	356	595	493	751	629	792	786	204	105	165	141	190	154	202	200	300	108	194	148	279	228	298	296	294	143	236	204	282	247	292	290
500	10	837	552	720	662	808	750	832	824	241	146	203	189	226	197	237	234	300	201	259	226	293	280	300	298	296	205	258	247	289	273	295	292
500	20	878	801	842	833	868	849	875	871	281	243	265	258	275	263	278	277	300	282	288	288	297	295	300	300	297	276	289	287	296	291	297	294
500	30	883	858	866	863	878	872	880	880	286	272	277	274	283	280	283	284	300	295	295	295	298	298	300	300	297	291	294	294	297	294	297	296
1024	0	765	153	387	279	655	464	758	750	186	27	83	60	152	91	185	184	296	95	186	137	269	219	294	292	283	31	118	82	234	154	279	274
1024	5	798	241	513	392	723	571	790	786	204	53	127	99	179	130	201	202	300	108	194	148	279	228	298	296	294	80	192	145	265	213	291	288
1024	10	837	542	718	656	794	744	831	825	241	141	204	187	222	196	237	235	300	201	259	226	293	280	300	298	296	200	255	243	279	268	294	292
1024	20	878	798	844	840	868	850	877	875	281	244	269	263	276	262	280	278	300	282	288	288	297	295	300	300	297	272	287	289	295	293	297	297
1024	30	883	859	867	866	880	875	882	882	286	272	278	275	285	282	285	285	300	295	295	295	298	298	300	300	297	292	294	296	297	295	297	297

Table C.2: Comparison of face detection algorithms (continued on the next page).

photo size	min face size	all photos: manual	all photos: opencvfront	all photos: opencvfrontalt	all photos: opencvfrontalt2	all photos: opencvfrontalttree	all photos: opencvprofile	all photos: coreimagelow	all photos: coreimagehigh	Flickr: manual	Flickr: opencvfront	Flickr: opencvfrontalt	Flickr: opencvfrontalt2	Flickr: opencvfrontalttree	Flickr: opencvprofile	Flickr: coreimagelow	Flickr: coreimagehigh	Geograph: manual	Geograph: opencvfront	Geograph: opencvfrontalt	Geograph: opencvfrontalt2	Geograph: opencvfrontalttree	Geograph: opencvprofile	Geograph: coreimagelow	Geograph: coreimagehigh	Panoramio: manual	Panoramio: opencvfront	Panoramio: opencvfrontalt	Panoramio: opencvfrontalt2	Panoramio: opencvfrontalttree	Panoramio: opencvprofile	Panoramio: coreimagelow	Panoramio: coreimagehigh
presence of faces																																
FP																																
240	0		72	31	48	6	29	2	1		27	15	21	1	12	2	1		17	6	12	1	6	0	0		28	10	15	4	11	0	0
240	5		78	31	50	7	30	2	1		29	15	22	1	12	2	1		17	6	12	1	6	0	0		32	10	16	5	12	0	0
240	10		92	38	57	11	38	8	7		43	22	29	5	21	8	7		16	6	12	1	5	0	0		33	10	16	5	12	0	0
240	20		49	16	32	6	18	2	1		26	8	18	2	13	2	1		9	4	6	1	2	0	0		14	4	8	3	3	0	0
240	30		18	7	14	4	4	1	2		11	3	9	1	4	1	2		1	1	2	0	0	0	0		6	3	3	3	0	0	0
500	0		439	206	308	51	171	6	15		91	37	58	11	47	1	4		201	110	159	27	77	2	4		147	59	91	13	47	3	7
500	5		442	203	305	47	169	6	12		99	39	63	14	50	2	4		192	106	152	21	72	2	4		151	58	90	12	47	2	4
500	10		284	116	174	28	86	4	12		94	37	51	14	43	3	6		99	41	74	7	20	0	2		91	38	49	7	23	1	4
500	20		76	35	44	9	28	2	6		37	15	22	5	17	2	3		18	12	12	3	5	0	0		21	8	10	1	6	0	3
500	30		24	16	19	4	10	2	2		13	8	11	2	5	2	1		5	5	5	2	2	0	0		6	3	3	0	3	0	1
1024	0		612	378	486	110	301	7	15		159	103	126	34	95	1	2		201	110	159	27	77	2	4		252	165	201	49	129	4	9
1024	5		557	285	406	75	227	8	12		151	77	105	25	74	3	2		192	106	152	21	72	2	4		214	102	149	29	81	3	6
1024	10		295	119	181	43	93	6	12		100	37	54	19	45	4	6		99	41	74	7	20	0	2		96	41	53	17	28	2	4
1024	20		80	34	38	10	28	1	3		37	12	18	5	19	1	3		18	12	12	3	5	0	0		25	10	8	2	4	0	0
1024	30		24	16	17	3	8	1	1		14	8	11	1	4	1	1		5	5	5	2	2	0	0		5	3	1	0	2	0	0
FN																																
240	0		70	80	72	94	94	81	80		57	63	56	78	77	64	63		4	4	4	4	4	4	4		9	13	12	12	13	13	13
240	5		43	47	41	62	62	48	47		41	45	39	60	59	46	45		0	0	0	0	0	0	0		2	2	2	2	3	2	2
240	10		19	20	18	33	33	21	20		18	19	17	32	32	20	19		0	0	0	0	0	0	0		1	1	1	1	1	1	1
240	20		4	6	6	9	6	4	4		4	6	6	9	6	4	4		0	0	0	0	0	0	0		0	0	0	0	0	0	0
240	30		5	4	4	5	4	4	4		5	4	4	5	4	4	4		0	0	0	0	0	0	0		0	0	0	0	0	0	0
500	0		28	45	40	68	64	60	58		22	35	33	55	50	44	43		1	1	0	2	2	4	3		5	9	7	11	12	12	12
500	5		13	27	24	44	41	31	29		13	25	22	42	38	29	27		0	0	0	0	0	0	0		0	2	2	2	3	2	2
500	10		14	15	15	27	25	18	17		13	14	14	26	24	17	16		0	0	0	0	0	0	0		1	1	1	1	1	1	1
500	20		6	4	5	9	7	3	3		6	4	5	9	7	3	3		0	0	0	0	0	0	0		0	0	0	0	0	0	0
500	30		5	4	4	5	4	2	3		5	4	4	5	4	2	3		0	0	0	0	0	0	0		0	0	0	0	0	0	0
1024	0		5	22	7	47	37	62	57		3	18	6	40	29	45	42		1	1	0	2	2	4	3		1	3	1	5	6	13	12
1024	5		5	23	11	41	31	33	30		4	21	10	39	29	31	28		0	0	0	0	0	0	0		1	2	1	2	2	2	2
1024	10		11	18	13	28	26	18	16		10	17	12	27	26	17	15		0	0	0	0	0	0	0		1	1	1	1	0	1	1
1024	20		6	5	4	8	6	5	4		6	5	4	8	6	5	4		0	0	0	0	0	0	0		0	0	0	0	0	0	0
1024	30		5	4	5	5	4	3	3		5	4	5	5	4	3	3		0	0	0	0	0	0	0		0	0	0	0	0	0	0
distance to ideal																																
240	0		100.4	85.8	86.5	94.2	98.4	81.0	80.0		63.1	64.8	59.8	78.0	77.9	64.0	63.0		17.5	7.2	12.6	4.1	7.2	4.0	4.0		29.4	16.4	19.2	12.6	17.0	13.0	13.0
240	5		89.1	56.3	64.7	62.4	68.9	48.0	47.0		50.2	47.4	44.8	60.0	60.2	46.0	45.0		17.0	6.0	12.0	1.0	6.0	0.0	0.0		32.1	10.2	16.1	5.4	12.4	2.0	2.0
240	10		93.9	42.9	59.8	34.8	50.3	22.5	21.2		46.6	29.1	33.6	32.4	38.3	21.5	20.2		16.0	6.0	12.0	1.0	5.0	0.0	0.0		33.0	10.0	16.0	5.1	12.0	1.0	1.0
240	20		49.2	17.1	32.6	10.8	19.0	4.5	4.1		26.3	10.0	19.0	9.2	14.3	4.5	4.1		9.0	4.0	6.0	1.0	2.0	0.0	0.0		14.0	4.0	8.0	3.0	3.0	0.0	0.0
240	30		18.7	8.1	14.6	6.4	5.7	4.1	4.5		12.1	5.0	9.8	5.1	5.7	4.1	4.5		1.0	1.0	2.0	0.0	0.0	0.0	0.0		6.0	3.0	3.0	3.0	0.0	0.0	0.0
500	0		439.9	210.9	310.6	85.0	182.6	60.3	59.9		93.6	50.9	66.7	56.1	68.6	44.0	43.2		201.0	110.0	159.0	27.1	77.0	4.5	5.0		147.1	59.7	91.3	17.0	48.5	12.4	13.9
500	5		442.2	204.8	305.9	64.4	173.9	31.6	31.4		99.8	46.3	66.7	44.3	62.8	29.1	27.3		192.0	106.0	152.0	21.0	72.0	2.0	4.0		151.0	58.0	90.0	12.2	47.1	2.8	4.5
500	10		284.3	117.0	174.6	38.9	89.6	18.4	20.8		94.9	39.6	52.9	29.5	49.2	17.3	17.1		99.0	41.0	74.0	7.0	20.0	0.0	2.0		91.0	38.0	49.0	7.1	23.0	1.4	4.1
500	20		76.2	35.2	44.3	12.7	28.9	3.6	6.7		37.5	15.5	22.6	10.3	18.4	3.6	4.2		18.0	12.0	12.0	3.0	5.0	0.0	0.0		21.0	8.0	10.0	1.0	6.0	0.0	3.0
500	30		24.5	16.5	19.4	6.4	10.8	2.8	3.6		13.9	8.9	11.7	5.4	6.4	2.8	3.2		5.0	5.0	5.0	2.0	2.0	0.0	0.0		6.0	3.0	3.0	0.0	3.0	0.0	1.0
1024	0		612.0	378.6	486.1	119.6	303.3	62.4	58.9		159.0	104.6	126.1	52.5	99.3	45.0	42.0		201.0	110.0	159.0	27.1	77.0	4.5	5.0		252.0	165.0	201.0	49.3	129.1	13.6	15.0
1024	5		557.0	285.9	406.1	85.5	229.1	34.0	32.3		151.1	79.8	105.5	46.3	79.5	31.1	28.1		192.0	106.0	152.0	21.0	72.0	2.0	4.0		214.0	102.0	149.0	29.1	81.0	3.6	6.3
1024	10		295.2	120.4	181.5	51.3	96.6	19.0	20.0		100.5	40.7	55.3	33.0	52.0	17.5	16.2		99.0	41.0	74.0	7.0	20.0	0.0	2.0		96.0	41.0	53.0	17.0	28.0	2.2	4.1
1024	20		80.2	34.4	38.2	12.8	28.6	5.1	5.0		37.5	13.0	18.4	9.4	19.9	5.1	5.0		18.0	12.0	12.0	3.0	5.0	0.0	0.0		25.0	10.0	8.0	2.0	4.0	0.0	0.0
1024	30		24.5	16.5	17.7	5.8	8.9	3.2	3.2		14.9	8.9	12.1	5.1	5.7	3.2	3.2		5.0	5.0	5.0	2.0	2.0	0.0	0.0		5.0	3.0	1.0	0.0	2.0	0.0	0.0

photo size	min face size	all photos								Flickr								Geograph								Panoramio							
		manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh	manual	opencvfront	opencvfrontalt	opencvfrontalt2	opencvfrontalttree	opencvprofile	coreimagelow	coreimagehigh
chi square with mode answer, question 5 (Are people the main subject of the photograph?)																																	
240	0	557.2	86.9	105.6	113.8	103.9	62.2	211.6	224.2	182.2	24.1	27.0	36.1	32.0	17.9	54.2	59.2	57.8	2.0	0.1	0.2	0.0	0.1	–	–	80.0	6.3	0.8	0.2	2.5	0.6	23.8	23.8
240	5	434.1	86.9	105.6	113.8	103.9	62.2	211.6	224.2	135.3	24.1	27.0	36.1	32.0	17.9	54.2	59.2	–	2.0	0.1	0.2	0.0	0.1	–	–	6.7	6.3	0.8	0.2	2.5	0.6	23.8	23.8
240	10	267.2	88.3	83.4	76.8	81.3	60.0	175.6	187.9	78.7	24.1	21.3	23.9	28.2	16.5	44.9	49.7	–	2.2	0.1	0.2	0.0	0.1	–	–	0.0	6.3	0.4	0.7	0.2	0.6	–	–
240	20	62.0	19.5	5.5	6.8	7.6	25.0	50.6	45.0	16.7	3.0	1.0	0.9	5.9	5.4	13.5	12.0	–	0.2	0.1	0.1	0.0	0.0	–	–	–	4.0	0.2	0.3	0.1	0.1	–	–
240	30	33.6	1.9	2.3	3.9	2.4	21.7	16.8	22.4	8.9	0.0	0.8	0.4	2.9	4.6	4.4	5.9	–	0.0	0.0	0.0	–	–	–	–	–	0.3	0.1	0.1	0.1	–	–	–
500	0	557.2	12.6	56.0	14.3	92.3	31.9	300.8	250.0	182.2	18.0	48.8	20.5	45.0	18.5	97.9	88.0	57.8	2.4	8.4	1.4	0.6	0.1	0.0	0.1	80.0	0.2	1.0	0.0	0.2	0.0	3.3	1.2
500	5	434.1	14.2	44.9	15.8	101.5	32.6	300.1	261.4	135.3	16.7	39.8	19.2	43.2	17.1	93.4	83.6	–	2.9	1.4	0.2	0.4	0.0	0.0	0.1	6.7	0.2	1.4	0.0	0.5	0.0	6.7	3.3
500	10	267.2	20.6	51.3	15.9	75.6	49.0	183.3	150.7	78.7	9.1	30.3	7.3	24.8	12.5	51.6	47.2	–	0.1	0.8	0.1	0.1	0.4	–	0.0	0.0	0.0	0.2	0.7	0.3	0.0	0.0	0.2
500	20	62.0	8.1	11.4	6.8	8.0	12.9	44.2	35.1	16.7	0.0	3.4	1.2	2.9	2.6	10.5	12.1	–	1.8	0.2	0.2	0.1	0.1	–	–	–	6.1	1.5	0.4	0.0	0.3	–	0.1
500	30	33.6	3.3	4.7	8.8	7.9	3.1	21.9	16.5	8.9	0.6	1.6	2.5	4.5	0.8	4.7	6.0	–	0.1	0.1	0.1	0.0	0.0	–	–	–	0.3	0.1	0.1	–	0.1	–	0.0
1024	0	557.2	13.5	43.3	39.4	105.3	38.9	292.1	270.4	182.2	1.6	15.6	22.2	42.0	9.1	94.7	98.1	57.8	2.4	8.4	1.4	0.6	0.1	0.0	0.1	80.0	1.5	3.0	2.4	7.4	4.2	3.3	5.9
1024	5	434.1	11.8	36.1	27.7	93.4	42.2	285.5	276.5	135.3	1.6	18.1	18.9	39.4	15.3	90.2	91.3	–	2.9	1.4	0.2	0.4	0.0	0.0	0.1	6.7	0.7	1.3	0.3	0.6	1.2	4.6	11.1
1024	10	267.2	30.9	43.7	25.9	59.5	46.6	181.5	171.5	78.7	12.2	28.2	13.1	20.6	12.2	52.9	55.4	–	0.1	0.8	0.1	0.1	0.4	–	0.0	0.0	1.8	0.1	2.0	0.2	0.7	0.1	0.2
1024	20	62.0	5.0	7.1	12.1	5.5	29.4	39.3	43.8	16.7	0.0	3.3	2.7	1.6	7.3	10.5	10.4	–	1.8	0.2	0.2	0.1	0.1	–	–	–	1.1	0.9	1.5	0.1	0.2	–	–
1024	30	33.6	0.5	4.7	8.0	3.8	15.7	22.4	22.4	8.9	0.4	1.6	1.6	2.9	8.9	5.9	5.9	–	0.1	0.1	0.1	0.0	0.0	–	–	–	0.2	0.1	0.0	–	0.1	–	–
chi square with mode answer (Does the photograph suggest this is a nice place to walk?)																																	
240	0	13.1	1.1	2.1	1.8	0.8	4.0	3.0	3.7	1.5	0.2	0.0	0.4	0.0	0.3	0.1	0.3	0.4	0.1	0.2	2.5	2.3	0.2	–	–	1.9	0.2	0.5	0.5	0.0	0.9	0.2	0.2
240	5	15.4	1.1	2.1	1.8	0.8	4.0	3.0	3.7	3.2	0.2	0.0	0.4	0.0	0.3	0.1	0.3	–	0.1	0.2	2.5	2.3	0.2	–	–	0.5	0.2	0.5	0.5	0.0	0.9	0.2	0.2
240	10	6.0	0.7	0.4	0.7	0.3	3.6	1.2	1.7	0.3	0.2	0.5	0.0	0.3	0.6	0.1	0.0	–	0.0	0.2	2.5	2.3	0.1	–	–	4.5	0.2	0.3	0.7	0.0	0.9	–	–
240	20	9.6	1.0	1.2	4.2	0.8	3.7	7.2	7.2	4.7	1.1	0.0	1.3	–	2.7	3.5	3.5	–	0.1	0.0	0.3	2.3	0.9	–	–	–	0.1	0.5	2.9	0.5	0.5	–	–
240	30	4.8	0.1	1.2	1.6	0.0	2.4	4.8	4.8	2.3	1.0	0.0	0.4	–	1.1	2.3	2.3	–	0.4	–	0.4	–	–	–	–	–	1.1	1.4	1.4	0.5	–	–	–
500	0	13.1	0.9	1.2	4.3	3.0	0.5	11.0	7.6	1.5	0.6	0.6	8.6	0.2	2.0	2.1	1.5	0.4	0.1	0.1	0.1	1.5	4.9	–	2.3	1.9	0.1	0.0	0.1	0.2	0.5	1.6	0.1
500	5	15.4	0.3	0.5	3.5	0.4	0.4	9.1	6.7	3.2	0.4	0.6	8.6	0.4	2.0	1.5	1.0	–	0.4	0.2	0.3	0.2	4.5	–	2.3	0.5	0.0	0.2	0.0	0.7	0.1	0.5	0.9
500	10	6.0	1.3	0.1	0.1	1.9	1.0	1.2	0.1	0.3	0.2	0.0	1.5	8.3	0.4	0.0	0.7	–	0.4	0.3	0.6	1.1	0.0	–	2.3	4.5	0.0	1.4	1.4	1.4	0.8	0.2	0.7
500	20	9.6	2.4	3.4	7.5	0.0	1.1	7.2	1.2	4.7	1.5	1.3	3.8	1.8	1.5	3.5	1.3	–	0.0	1.8	2.7	1.9	0.3	–	–	–	0.1	1.4	0.2	0.2	0.0	–	0.5
500	30	4.8	0.5	5.2	8.2	0.4	0.6	4.8	2.0	2.3	0.3	1.8	4.5	–	2.3	2.3	2.3	–	0.7	2.1	2.1	0.4	0.9	–	–	–	1.1	0.5	0.5	–	0.5	–	0.2
1024	0	13.1	0.1	0.0	0.2	2.1	0.1	13.5	11.2	1.5	0.3	0.0	4.5	0.1	1.2	1.5	2.1	0.4	0.1	0.1	0.1	1.5	4.9	–	2.3	1.9	0.1	0.1	1.3	0.2	0.0	8.7	1.4
1024	5	15.4	0.0	0.8	0.4	1.3	0.0	10.2	6.7	3.2	0.0	0.1	0.4	0.1	0.3	1.0	1.0	–	0.4	0.2	0.3	0.2	4.5	–	2.3	0.5	0.1	0.3	0.1	0.6	0.6	4.7	0.0
1024	10	6.0	1.9	0.1	0.1	0.5	5.5	2.8	1.5	0.3	0.0	0.9	0.3	5.7	2.6	0.0	0.3	–	0.4	0.3	0.6	1.1	0.0	–	2.3	4.5	0.1	0.8	0.0	0.0	1.4	4.5	0.5
1024	20	9.6	1.6	0.3	5.1	0.2	2.3	4.8	2.0	4.7	1.5	0.0	2.1	0.8	1.4	2.3	0.5	–	0.0	1.8	2.7	1.9	0.3	–	–	–	0.2	2.1	0.2	0.2	0.1	–	–
1024	30	4.8	1.6	2.8	8.0	0.4	0.3	4.8	4.8	2.3	0.8	0.3	3.4	–	1.1	2.3	2.3	–	0.7	2.1	2.1	0.4	0.9	–	–	–	0.9	0.5	0.2	–	1.4	–	–

Table C.3: Matching of answers by survey respondents with results of face detection.

Appendix D

Papers, presentations and media

Kachkaev, A. and Wood, J. 2012. "Using Visual Analytics to Detect Problems in Datasets Collected From Photo-Sharing Services." Poster presented at the *IEEE Conference on Information Visualization (InfoVis)*, 14 - 19 Oct 2012, Seattle, Washington, US.
`http://openaccess.city.ac.uk/1320/`

Kachkaev, A. and Wood, J. 2013. "Crowd-sourced Photographic Content for Urban Recreational Route Planning." Paper presented at the *45th Annual Universities' Transport Study Group Conference*, 2 - 4 Jan 2013, Oxford, UK.
`http://openaccess.city.ac.uk/2828/`

Kachkaev, A. and Wood, J. 2013. "Investigating Spatial Patterns in User-Generated Photographic Datasets by Means of Interactive Visual Analytics." Paper presented at *GeoViz Hamburg: Interactive Maps that Help People Think*, 6 - 8 Mar 2013, HafenCity University, Hamburg, Germany.
`http://openaccess.city.ac.uk/2829/`

Kachkaev, A. and Wood, J. 2013. "Exploring Subjective Survey Classification of a Photographic Archive using Visual Analytics." Poster presented at the *IEEE Conference on Information Visualization (IEEE VIS 2013)*, 13 - 18 Oct 2013, Atlanta, Georgia, US.
`http://openaccess.city.ac.uk/2785/`

Kachkaev, A. and Wood, J. 2014 "Automated planning of leisure walks based on crowd-sourced photographic content Planning." Paper presented at the *46th Annual Universities' Transport Study Group Conference*, 6 - 8 Jan 2014, Newcatle, UK.
`http://openaccess.city.ac.uk/4943/`

Kachkaev, A., Wood, J. and Dykes, J. 2014. "Glyphs for Exploring Crowd-sourced Subjective Survey Classification." *Computer Graphics Forum* 33(3):311–320.
`http://openaccess.city.ac.uk/3393/`

⋆ *best paper award at the Eurographics Conference on Visualization (EuroVis), 9-13 Jun 2014, Swansea, Wales, UK*

Selected talk:

"Leisure routing for pedestrians based on the wisdom of the crowd."
Three Minute Thesis (3MT®) competition, 27 Mar 2014, City University London, UK.

⋆ *first prize, invitation to UK semi-final at York University*

Maps representing the values of luminance in photographic datasets (see Subsection 4.5.2 on page 189) have been mentioned in some online media:

2014-06-24. “Which Bits Of London Are Most Photographed?”
Londonist (by Matt Brown)
`http://londonist.com/2014/06/which-bits-of-london-are-most-photographed`

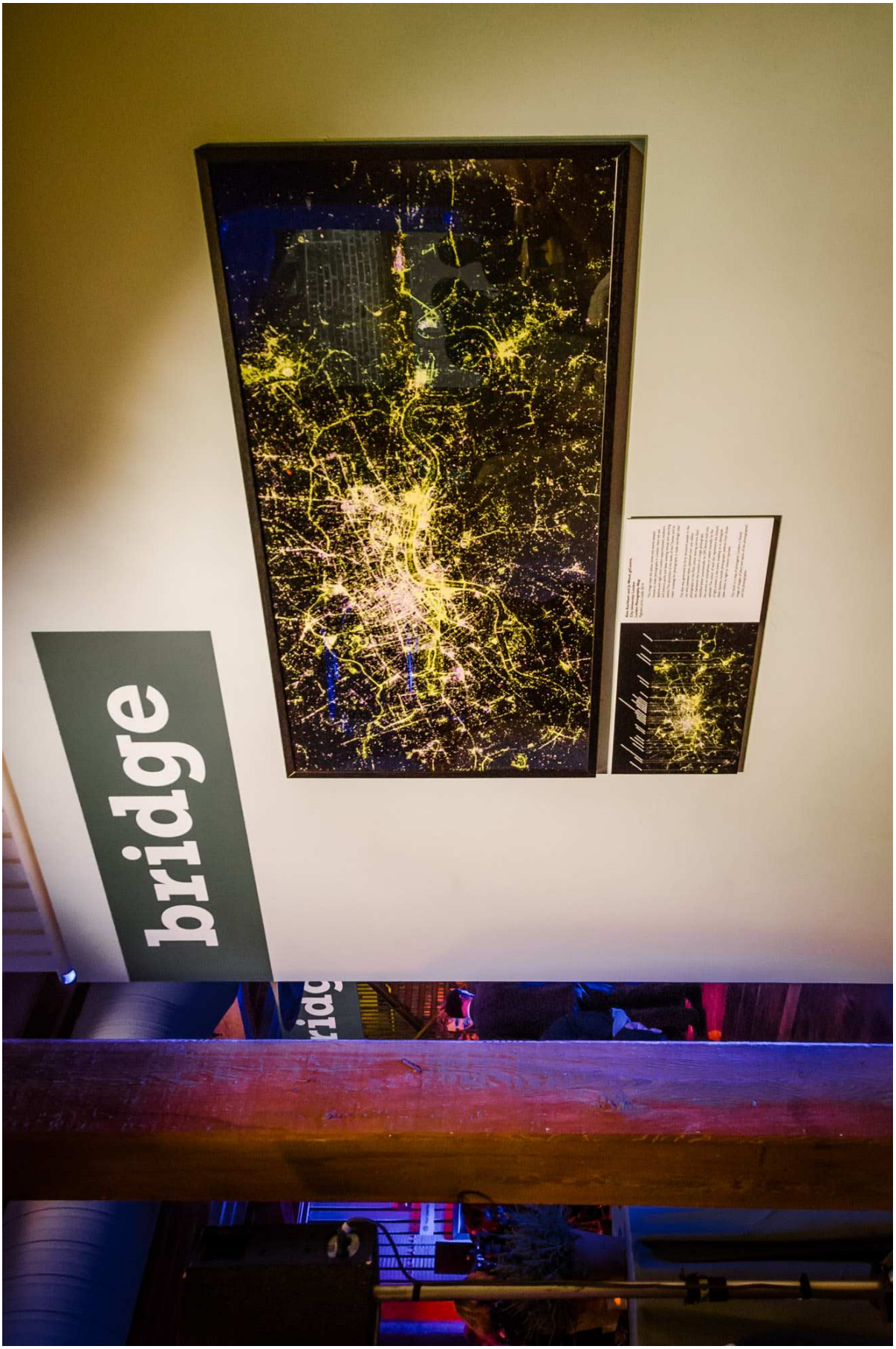
2014-06-27. “Tracking the most photographed locations in London.”
City University London (by John Stevenson)
`http://bit.ly/cityacuk_luminance_map`

2014-07-11. “Data map reveals Regent’s Canal is among most snapped places in London on social media.”
Camden New Journal (by Alina Polianskaya)
`http://bit.ly/camdennewjournal_luminance_map`

2014-10-22. “London’s most photogenic locations.”
Telegraph (by Oliver Uberti)
`http://bit.ly/telegraph_luminance_map`

2014-11-12. “Twelve data maps that sum up London.”
BBC News Magazine (by Paul Kerley)
`http://www.bbc.co.uk/news/magazine-29915801`

Flickr luminance map (Figure 4.58 on page 199) has been included into a book titled *“LONDON: The Information Capital: 100 maps and graphics that will change how you view the city”* by James Cheshire and Oliver Uberti (2014). This visualization was also displayed in the Museum of London Docklands during *“Bridge”* exhibition, which was held between July and November 2014.



Flickr luminance map of London at the Museum of London Docklands (26 July 2014)