EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

Automatic Generation of Tourist Brochures

Michael Birsak [1] Przemyslaw Musialski [1] Peter Wonka [2,3] Michael Wimmer [1]

[1]Vienna University of Technology, Austria [2]KAUST, Saudi Arabia [3]Arizona State University, USA

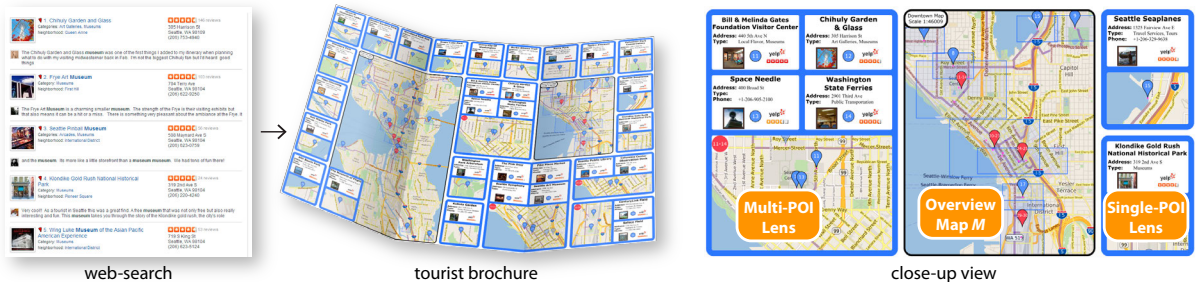


Figure 1: *Automatic generation of tourist brochures on demand: the user selects a set of points of interest (POIs) obtained from a free web database in a particular region (e.g., City of Seattle). Our system generates a comprehensive but also detailed tourist brochure that provides walking or driving directions, detail lenses, and short descriptions of the selected POIs.*

Abstract

We present a novel framework for the automatic generation of tourist brochures that include routing instructions and additional information presented in the form of so-called detail lenses. The first contribution of this paper is the automatic creation of layouts for the brochures. Our approach is based on the minimization of an energy function that combines multiple goals: positioning of the lenses as close as possible to the corresponding region shown in an overview map, keeping the number of lenses low, and an efficient numbering of the lenses. The second contribution is a route-aware simplification of the graph of streets used for traveling between the points of interest (POIs). This is done by reducing the graph consisting of all shortest paths through the minimization of an energy function. The output is a subset of street segments that enable traveling between all the POIs without considerable detours, while at the same time guaranteeing a clutter-free visualization.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Travel brochures are special maps that allow a user to find out about points of interest (POI) in an area, and how to get to those POIs. Classical travel brochures are part of travel guides, available as handouts in hotels, or found as posters in public areas such as subway stations. The POIs are typically highlighted (by using a different color or visual representation), and have numbered cross-references to a separate list or description of the POIs. Classical travel brochures suffer from several problems: (1) it is tedious to find information about nearby POIs because they have to be cross-referenced into an arbitrarily sorted list, (2) they are not suitable for larger areas because the map level-of-detail will be too low near important areas, e.g., near the POIs, and (3) they do not offer guidance for efficient travel from one POI to another.

Problems (1) and (2) lead to a *layout problem*: how can the POIs and detail information about them be laid out on a map so as to allow efficient cross-referencing between the map and the detail, even for larger areas? Problem (3) leads to a *routing problem*: how is it possible to represent on a single map efficient routes between multiple POIs? Previous research on route visualization has focused on single route maps (from point A to point B) and on destination maps (from anywhere to a point A). Here, however, we have to solve the problem how to get from any POI to any other POI, which is a new route visualization problem.

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

This is the authors preprint. The definitive version of the paper is available at http://diglib.eg.org and http://onlinelibrary.wiley.com

In this paper, we present a novel framework for generating tourist brochures for a given set of POIs that contributes solutions to these problems:

- Detail about POIs is provided by so-called *detail lenses*, which contain a high-resolution map cut-out for one or more POIs, together with auxiliary information like name, address, or a photo. Detail lenses address problem (2) and are discussed in Section 4. We also explain how to cluster lenses so as to avoid clutter and overlapping POI markers in the overview map.
- Easy cross-referencing between the overview map and the detail lenses is facilitated by arranging the detail lenses around the overview map using an automatic layout algorithm that optimizes spatial proximity between a lens and its corresponding marker on the map. This algorithm addresses problem (1) and is discussed in Section 5.
- Finally, we calculate efficient paths between all POIs, solving a multi-route problem and the attached route visualization problem. The challenge is to simplify the route graph so as to provide legibility while maintaining route efficiency. This addresses problem (3), and is discussed in Section 6.

Figure 1 (center) shows an automatically generated tourist brochure for Seattle, and a closeup of the particular elements (right). We evaluate the components of the framework with user studies and quantitative measurements in Section 7.

2. Related Work

There are three main steps involved in planning a trip: (1) the general decision-making step, in which the destination of the trip is chosen, (2) the information-acquisition step, where data about the way and the destination is acquired, and finally (3) the detailed decision making, where the precise schedule of the trip is prepared [Zal96, FM98, SMSW90]. The last two steps require quite an effort from the traveler, and often turn out as difficult tasks that are hard to solve optimally. The introduction of the Internet for everyday usage had a significant impact on trip planners, who discovered that medium as an important source of information. This influence has been studied [LFC04, PF06] and shown that it has a significant impact on the choice of the destinations and routes.

Furthermore, the availability of online geographic maps combined with other information databases brought about various online tools with the aim to help the traveler to plan the trip. Shiraishi et al. [SSN*05] presented a personal planner for mobile devices that helps to prepare a trip schedule with multiple destinations under the constraints of time and user preferences. Agrawala et al. [AS01] created route maps that are similar to hand-drawn sketches. Grabler et al. [GASP08] proposed a system for the automatic generation of tourist maps that simplify the overall visual complexity and reduce the appearance to a small number of significant and easily recognizable landmarks placed on a simplified street network. Kopf et al. [KAB*10] proposed a system for selecting and laying out the important roads based on mental representations of road networks. Crandall et al. [CBHK09] proposed a system to analyze geotagged Internet photo collections and to lay them out on schematic world maps. Karnick et al. [KCJ*10] introduced a system to visualize routes using local *detail lenses* automatically placed on the canvas. In this work we continue the idea of detail lenses that are used to amplify particular regions of the map. Most recently, Zheng et al. [ZYZ*13] proposed a system for trip planning along routes with sightseeing qualities. Their system produces routes from A to B that are enriched by nearby POIs. In contrast, our approach produces a route graph for multiple POIs without any specific direction. Another related work deals with the problems of effective routing. Bast et al. [BFSS07] provide a method to reduce quickest path queries in road networks to a small number of table lookups, and Sanders and Schulte [SS07] outline algorithms for transportation and routing problems beyond Dijkstra's shortest-path method.

Finally, our work is related to methods that deal with layouts, which in general refers to the process of determining the size and position of the visual objects in an information presentation. Lok et al. [LFN04] introduce a WeightMap to control automated layouts. Jacobs et al. [JLS*03] provide a system for grid-based document layout generation, and Di Iorio [DFV*08] proposed to generate layouts by describing their topological properties rather than geometric ones. Chen et al. [CBGS09] introduced a method for automatic generation of layouts in Mangas. Layouts received attention not only in connection to documents and maps: Merrell et al. [MSL*11] proposed a method for discrete layouts of buildings, and Yeh et al. [YYW*12] introduced a system for the synthesis of layouts based on probabilistic optimization.

3. Design Principles

To accomplish automatic design tasks, Agrawala et al. [ALB11] proposed to formulate design principles which serve as guidelines for algorithmic solutions. For our purposes, we identified four design principles that should be fulfilled as best as possible to produce a tourist brochure. In that sense, the principles can also be viewed as evaluation criteria by which the effectiveness of our design can be evaluated [ALB11], which we do in Section 7.

Spatial Grouping (P1): The first design principle is the idea of spatial grouping: in the case when the points of interest exhibit very dense clusters, or in other words, when the pins on the overview map become too close and cannot be rendered without overlap, they should be grouped. We do this using multi-POI lenses, which are represented by their own map pin.

Spatial Proximity (P2): The second principle relies on the observation that a layout is easier to understand if coherent pieces of information are placed in proximity of one another and are not scattered across the page. In partic-

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

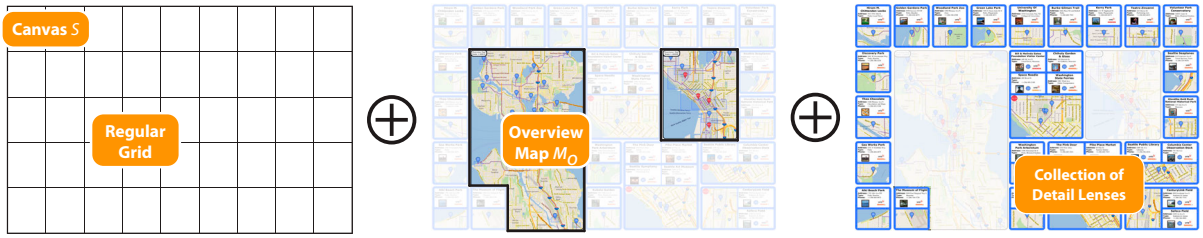


Figure 2: *The canvas of the tourist brochure is divided into cells of a regular grid. The content is composed of two basic elements: maps and detail lenses. All basic elements are placed in the cells of the regular grid.*

ular, lenses and their corresponding map pins should be placed as near as possible to each other.

Arrangement Similarity (P3): The third design principle states that the spatial arrangement of the map pins should resemble as best as possible the arrangement of the respective lenses on the brochure page. We assume the user will find the correspondences more intuitively if both arrangements are similar.

Route Simplicity (P4): The fourth design principle is the clarity of the routing graph. A major goal of our customized brochures is to provide the user routing directions from any point to any other point in a manner that is easy to follow, so the graph should not be too dense and too cluttered.

4. Tourist Brochures with Maps and Detail Lenses

The brochures are composed of top-view maps with routing information and a number of detail lenses, which emphasize particular points of interest.

4.1. Overview of Brochure Elements

All elements are placed on a rectangular 2d domain with user-specified format (e.g., DIN A3 landscape), which we call *canvas S*. In order to align the elements on S, we employ a *grid G*—a canonical page design pattern. In our case, the grid is regular and subdivides the canvas into a set of equally sized cells with integer coordinates. The grid is filled with three kinds of elements (Figure 2):

Map Cells: elements that form overview maps. We distinguish between the *main overview map* M_O and *optional overview maps* $M_1,\ldots,M_{n_M}$, which are at a higher zoom level and cover subregions of M_O. Each map is composed of a 4-connected set of tiles that does not have to be convex. In practice, it is usually a rectangular or L-shaped region, although also U-, H-, or other shapes may occur.

Lens Cells: a lens is a unit of additional information that describes a particular site. We distinguish between *single-POI* and *multi-POI* lenses. A single-POI lens contains information about exactly one point of interest, while a multi-POI lens contains multiple POIs combined into one lens. Our system groups POIs automatically depending on their spatial location in the main overview map.

Empty Cells: grid cells which have not been assigned any information tiles remain empty. Our system fills them with random photographs, but they could be used to display advertisements or additional information.

4.2. Points of Interest

The high-level input to our system is a set of POIs, for example selected from a public online tourist database. In a set $\boldsymbol{P} = \{P_1, P_2, \ldots, P_{n_P}\}$ of n_P POIs, each P_i encapsulates the following attributes:

- $Pos(P_i)$ is the Mercator-projected position of P_i with each component in meters,
- $Rect(P_i)$ is the geographic bounding box of P_i, which is the minimum actual earth region that should be shown in a detail lens for P_i. The size of P_i is uniquely defined by the size of a single cell in the layout grid G and a given zoom level.

The geographical region covered by the POIs can be of various size, e.g., a city, a region, or even an entire country.

4.3. Maps

The overview map M_O is positioned on S in compliance with the grid layout. Its geographic boundary is given by the bounding box of the boundaries of all POIs, so that

$$Rect(M_O) = Rect\left(\bigcup_{i=1}^{n_P} Rect(P_i)\right).$$

The position and size of M_O on S is determined interactively by the user. Optional map(s) of user-defined regions can also be positioned interactively on S in compliance with the grid layout, such that $\boldsymbol{M} = \{M_0 := M_O, M_1, M_2, \ldots, M_{n_M}\}$ is the set of $n_M + 1$ maps (overview map and maps of all user-defined regions) on S, where each M_i encapsulates the following attributes:

- $Rect(M_i)$ is the geographic bounding box defining the exact region on earth that is shown in M_i
- $Scale(M_i)$ is the scale denominator of the map M_i.

Map Scales. For simplicity, our tourist brochures use at most three different map scales. We have one map scale s_{M_0} for the main overview map, one map scale s_{M_1} for all optional overview maps and one map scale s_D for all the detail lenses. The exact values for the map scales can differ between different tourist brochures, but they always have to meet the following requirement: $s_{M_0} < s_{M_1} < s_D$.

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

To make things even simpler, we do not use arbitrary map scales, but fixed zoom levels as they are common in web-based mapping tools like Google Maps. A zoom level of 0, for example, maps the whole equator of the earth onto a small image with a side length of 256 pixels. Assuming a screen pixel density of 90.71 DPI, which results from a pixel width of 0.28 mm, a zoom level of 0 corresponds to a scale denominator of 559,082,264.03. An incrementing of the zoom level by 1 then corresponds to a bisection of the scale denominator. In our tests, we often found the zoom levels 12, 13 and 14 to perfectly fit our needs for the zoom levels in the main overview map, optional overview maps and detail lenses respectively. Zoom level 12, for example, corresponds to a scale denominator of 136,494.69.

Our system not only allows for standard screen pixel density of 90.71 DPI but for an arbitrary value. For high-quality print, one could also choose a value of 300.0. If a zoom level (and therefore also the scale denominator sd) for a particular map is chosen, the corresponding width in pixels w_p can easily be calculated by

$$w_p = w_m \cdot \frac{\text{DPI}}{0.0254 \cdot sd} ,$$

where w_m is the width in the Mercator projection.

4.4. Detail Lenses

Detail lenses are elements that contain one or more POIs and a map that covers the union of their bounding boxes. We define the set of detail lenses as $\boldsymbol{L} = \{L_1, L_2, \ldots, L_l\}$, where each lens L_i encapsulates a set of attributes:

- $\bar{\boldsymbol{P}}(L_i)$ is the set of all P_i contained in this lens ($\bar{\boldsymbol{P}} \subseteq \boldsymbol{P}$),
- $Rect(L_i)$ is the bounding box of the union of all $Rect(P_j)$ of $P_j \in \bar{\boldsymbol{P}}(L_i)$,
- $Pos(L_i)$ is the center of $Rect(L_i)$,
- $Scale(L_i)$ is the scale denominator of the map of L_i.

Detail lenses are created and placed automatically according to the input points of interest. To accomplish this, we introduce a lens creation process based on unsupervised clustering of the spatial locations of the POIs. Following design principle **(P1)**, we group the POIs which are located very close to each other to form multi-POI lenses. Such lenses are represented on the overview map by only one common symbol (cf. Fig. 1, right).

POI Grouping. For the clustering, we utilize the k-means algorithm on the input set $\boldsymbol{P}$ of POIs, where we use their spatial location on the map as a Euclidean 2d point. This step divides the POIs into a set of clusters $\boldsymbol{C} = \{C_1, C_2, \ldots, C_l\}$ with various numbers of P_j per C_i. Each cluster C_i corresponds to exactly one detail lens in the brochure. We use an iterative approach to determine the optimal number of clusters k.

We start with $k = 1$ for k-means and run the lens placement algorithm (see Section 5) to find a valid layout. Since $k = 1$ results in only one cluster and thus one big multi-lens containing all POIs, it is usually impossible to find a region in the grid to place it. Therefore, we iteratively increment k and run the placement algorithm until we find a valid layout. If no such an arrangement can be found even if the number of clusters is equal to the number of POIs, i.e., only single-POI lenses are present, then there exists no solution for the current set of overview maps and POIs. In this case, the user has to choose a bigger grid, fewer POIs or smaller/fewer overview maps.

Single-POI. In the easiest case, a cluster C_i consists of just one POI and the corresponding lens is therefore a single-POI lens, which occupies per definition exactly one grid cell. We use one half of such a lens to display an *information block* that contains the name and some further data like address, type and phone number. To give the user an idea of how the actual location looks like, a photo is provided if available. Furthermore, a rating value based on user comments in the online database is displayed inside the information block. The second half of the single-POI lens is covered by a detailed map showing the surrounding region centered around the actual POI.

Multi-POI. If the number of POIs in the cluster C_i is bigger than 1, we create a multi-POI lens that groups a number of points of interest. We start by calculating the size of the map, that shows the region corresponding to the bounding box of all the POIs of the lens. Then we identify the minimum number of grid cells in horizontal and vertical direction needed to display the map according to the zoom-level and canvas resolution. Usually, with this approach we end up with a layout of grid cells where we have not enough free half-cells that are untouched by the map to place the information blocks for all of the POIs of C_i. Therefore, we expand the lens until there is enough space to place all information blocks. We use a simple iterative method to find a suitable size: In every iteration, we expand the lens into the direction that corresponds to the axis with the smaller number of cells.

The layout optimization (cf. Section 5) as well as the calculation of the routing graph (cf. Section 6) are accomplished fully automatically, and in the following we describe the details of both components.

5. Automatic Layout Optimization

To *lay out* refers to the process of positioning and arranging contentual entities on a canvas. From our point of view, the main challenge is the design of dynamic layouts, i.e., layouts with a number of elements that varies depending on the content, as in the case of custom brochures. Thus, we formulate it as an optimization task.

Cost Function. We want to place the lenses in compliance with design principles **(P2)** and **(P3)** defined in Section 3. In order to enforce both of them, we formulate a global cost function with two corresponding terms, e_{prox} (for **(P2)**) and e_{sim} (for **(P2)**). The cost for placing lens L_k at position (i, j) in S is defined as:

$$f_{i,j,k} = w_k \left(\lambda_1 e_{prox} + \lambda_2 e_{sim}\right) , \qquad (1)$$

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

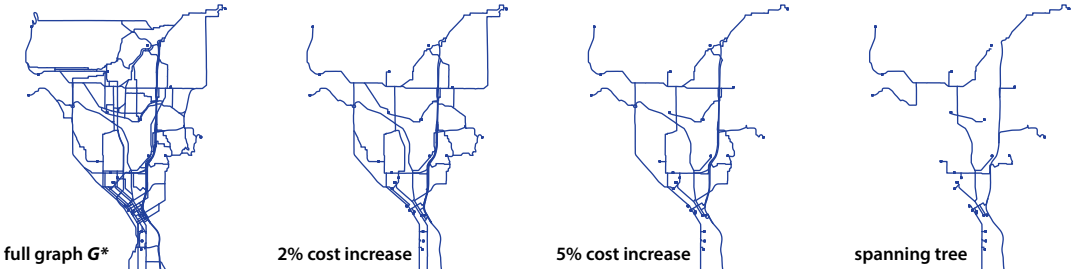


Figure 3: *Different steps of the graph simplification procedure described in Section 6. The first graph on the left is the union of all shortest paths. Consecutive graphs have been created by removing the most redundant segments. The final graph is a spanning tree that cannot be simplified anymore.*

with

$$i = 1,2,...,m;\ j = 1,2,...,n;\ k = 1,2,...,l\,,$$

and:

$$e_{prox} = \tfrac{1}{2}\|p^{\tilde{S}}_{mc}(L_k,i,j) - p^{\tilde{S}}_{M_{max}}(L_k)\|^2\,,$$

$$e_{sim} = \tfrac{1}{\pi}\left(\cos^{-1}(e_{ang})\right) + \tfrac{1}{2}e_{dist}\,,$$

$$e_{ang} = \frac{p^{\bar{M}_O}(L_k)\cdot p^{\bar{S}}_{mc}(L_k,i,j)}{\|p^{\bar{M}_O}(L_k)\|\cdot\|p^{\bar{S}}_{mc}(L_k,i,j)\|}\,,$$

$$e_{dist} = \tfrac{1}{\sqrt{2}}|\|p^{\bar{M}_O}(L_k)\| - \|p^{\bar{S}}_{mc}(L_k,i,j)\||\,,$$

where $w_k = |\bar{\boldsymbol{P}}(L_k)|$ is the number of POIs in the lens L_k, which scales the cost according to the importance of the lens.

In term e_{prox}, $\tilde{S}$ is the uniformly scaled canvas S such that its width w_S and its height h_S are both divided by $\max(w_S,h_S)$. The variables $p^{\tilde{S}}_{mc}(L_k,i,j)$ and $p^{\tilde{S}}_{M_{max}}(L_k)$ refer to the position of the map center of the lens L_k when positioned at grid cell (i,j), and to the position of the corresponding lens marker in the overview map with the highest zoom factor respectively, both with respect to $\tilde{S}$. The term e_{prox} is therefore half the squared distance between the map center of the lens L_k and the corresponding marker. We measure e_{prox} with respect to $\tilde{S}$ for normalization reasons, such that the squared distance can be at most 2. By dividing by 2, our term e_{prox} is in the range of 0 and 1.

Term e_{sim} in turn consists of two terms. Both terms are a measure of how similar the lens L_k is positioned in the grid compared to the corresponding marker in the overview map M_O. The first term does this by comparing the angles, the second term by comparing the distances. All measurements are done with respect to $\bar{M}_O$, which is the non-uniformly scaled overview map M_O such that $w_{\bar{M}_O} = 1.0$ and $h_{\bar{M}_O} = 1.0$, and with respect to $\bar{S}$, where $w_{\bar{S}} = 1.0$ and $h_{\bar{S}} = 1.0$. The variables $p^{\bar{M}_O}(L_k)$ and $p^{\bar{S}}_{mc}(L_k,i,j)$ are then the position of the corresponding marker of L_k with respect to $\bar{M}_O$ and the position of the map center of the lens L_k positioned at grid cell (i,j) with respect to $\bar{S}$, respectively.

Our cost function aims at the minimization of both terms: the sum of squared distances of all lenses (measured at their respective local map pin) to their corresponding pins in the nearest overview map, and the measure for the similarity of the position of the lenses in the grid compared to the position of the markers in the overview map. In order to place the lenses in an optimal manner, we formulate the problem as a binary integer program (BIP) with the objective function:

$$E = \min_{x_{i,j,k}} \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{i,j,k}x_{i,j,k}\,, \tag{2}$$

where m is the number of rows in the grid, n is the number of columns in the grid, l is the number of lenses, and $f_{i,j,k}$ is the cost to place the lens L_k at grid cell (i,j). We introduce $m\cdot n\cdot l$ binary variables of the form $x_{i,j,k}$ and define that $x_{i,j,k} = 1$ when the lens L_k is placed (always with respect to its upper left cell) at grid cell (i,j), and $x_{i,j,k} = 0$ otherwise.

Hard Constraints. The objective is solved subject to the following constraints:

C1: Any lens L_k is positioned exactly once:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{i,j,k} = 1;\ k = 1,2,...,l\,. \tag{3}$$

The l constraints of type **C1** ensure that each lens L_k is positioned exactly once. We can achieve this goal by setting only one of the variables in the set of variables corresponding to L_k to 1, and all the others to 0. The sum of all variables corresponding to one particular lens has therefore to be 1.

C2: Lenses must be contained entirely inside the grid in X and Y:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} j\cdot x_{i,j,k} \le n - W_k + 1;\ k = 1,2,...,l\,, \tag{4}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} i\cdot x_{i,j,k} \le m - H_k + 1;\ k = 1,2,...,l\,, \tag{5}$$

where W_k and H_k refer to the number of grid cells occupied by the lens L_k on the discrete grid. The $2\cdot l$ constraints of type **C2** guarantee that no lens is (partially) out-

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

side of the grid in the horizontal (Eq. 4) or vertical (Eq. 5) direction. Although we only have the variables $x_{i,j,k}$ for $j = 1,2,...,n$ in the X-direction, therefore only for existing columns, a lens can be partially outside when, for example, $W_{k_1} = 2$ for a particular lens L_{k_1} and $x_{i,n,k_1} = 1$ for any $i \in \{1,2,...,m\}$, since the lenses are placed with respect to their upper left cell. The sum in the constraint gives us the column in which the particular lens L_k is placed. Note that the sum only gives us the correct column if the lens is positioned only once, which is already ensured by **C1**. When we know the column c_L of the upper left cell, the remaining task is to guarantee that $c_L \leq n - W_k + 1$. The same applies in the vertical direction, where in row r_L, we have to guarantee that $r_L \leq m - H_k + 1$.

C3: No more than one lens is placed at any grid cell:

$$\sum_{k=1}^{l} \sum_{a=\max(1,i-H_k+1)}^{i} \sum_{b=\max(1,j-W_k+1)}^{j} x_{a,b,k} \leq 1 \qquad i = 1,2,...,m;\ j = 1,2,...,n\,. \tag{6}$$

The $m \cdot n$ constraints of type **C3** ensure that no more than one lens occupies any grid cell. Unfortunately, it is not sufficient to only test every grid cell (i, j) if $x_{i,j,k}$ is 1 for more than one $k \in \{1,2,...,l\}$, since lenses bigger than 1×1 can occupy a grid cell also when they are positioned in a neighbor cell. The smallest row number where a lens L_k positioned there would still range to the row i can easily be computed by $i - H_k + 1$. Analogous, the smallest column number can be computed by $j - W_k + 1$. In order to ensure that only one lens occupies a grid cell (i, j), we therefore have to sum over all lenses, and for every lens, over the whole region in which the lens would affect the grid cell. The *max* functions in the constraint are just needed to avoid access of a grid cell index smaller than 1.

C4: Map areas must not be occupied by lenses:

$$\sum_{k=1}^{l} \sum_{a=\max(1,i-H_k+1)}^{i} \sum_{b=\max(1,j-W_k+1)}^{j} x_{a,b,k} = 0 \qquad (i,j) \in I\,, \tag{7}$$

where I is the set of all indices (i, j) occupied by an overview map. The constraints of type **C4** are very similar to the ones of **C3**. They ensure that grid cells where overview maps are placed are not occupied by lenses. The maximum number of lenses in such grid cells has therefore to be 0. There exists one constraint **C4** for every index.

We solve this optimization problem using the Gurobi-Library [Gur13], which leverages a linear-programming based branch-and-bound algorithm to find a good feasible solution. The resulting layout ensures that, given the specific input, all lenses are placed in the best possible neighborhood of their corresponding map pins. Finally, to make the brochure even easier to read, we perform a row-wise re-enumeration of the lenses and their corresponding POIs in a top-to-bottom and left-to-right scheme.

6. Routing Graph

Our final design principle **(P4)** demands routing information that is easy to follow and in some sense "optimal", i.e., the route from any point to any other should be efficient (cf. Section 3). The most efficient solution would connect each pair of POIs with their shortest path. But this would result in a considerable number of overlapping paths. While a single route from A to B can be rendered in a very recognizable way, a whole network of multiple paths becomes cluttered quickly as the number of segments increases. Therefore, we propose to simplify the network graph considerably, and to leave only a small subset of the original segments, which still provides fast (but not necessarily the fastest) connections from any point to any other. Note that the requirements are contradictory, and the task is a balancing task.

The formal goal is to find optimal routes between all POIs, such that it is possible to reach each POI from any other POI. This computation is carried out on a directed street-graph $\boldsymbol{G}$ obtained from a geographical information system (GIS).

Graph $\boldsymbol{G}^*$. We define the graph of all streets in the overview map M_O as a tuple $\boldsymbol{G} = (\boldsymbol{N},\boldsymbol{A})$, where $\boldsymbol{N}$ is the set of all nodes in $\boldsymbol{G}$, with $\boldsymbol{P} \subset \boldsymbol{N}$, and $\boldsymbol{A}$ is the set of all arcs connecting the nodes. In that graph we compute the set of all shortest paths between all pairs (P_i, P_j) of $\boldsymbol{P}$ using Dijkstra's shortest-path algorithm. We denote the shortest paths as R^*_{ij} and their set as $\boldsymbol{R}^*$. This set serves as a starting point for another graph $\boldsymbol{G}^* = (\boldsymbol{N}^*,\boldsymbol{A}^*)$, which is composed of only a subset of all nodes and arcs of the full graph $\boldsymbol{G}$. To create $\boldsymbol{G}^*$, we find all crossings, branches and intersections of the shortest paths in $\boldsymbol{R}^*$, which we denote as the set $\boldsymbol{K}$. Together with the input POIs, they define the nodes of $\boldsymbol{G}^*$, i.e., $\boldsymbol{N}^* = \boldsymbol{K} \cup \boldsymbol{P}$. The set of edges $\boldsymbol{A}^*$ of $\boldsymbol{G}^*$ contains all sequences of arcs of $\boldsymbol{A}$ that connect nodes $\boldsymbol{N}^*$ and lie on the shortest paths $\boldsymbol{R}^*$. We denote the elements of $\boldsymbol{A}^*$ as sections s_i. In other words, $\boldsymbol{G}^*$ is the graph defined by all shortest paths between all POIs, as depicted in Figure 3, left hand side.

Route-Aware Graph Simplification. The resulting graph basically gives us routing directions how to reach from each POI all other POIs. Unfortunately, the graph is usually quite dense and ambiguous, which makes it difficult to grasp intuitively. The ideal routing graph should be as sparse as possible, or even just a spanning tree. In order to simplify $\boldsymbol{G}^*$, we propose a greedy iterative procedure which pursues two goals: keeping the driving/walking time as low as possible while at the same time removing as many sections as possible. Note that these goals are contradictory by their nature, since the removal of sections leads to an increase of driving/walking time. During our approach, we successively reduce $\boldsymbol{G}^*$ and denote our residual graph $\bar{\boldsymbol{G}}^* = (\bar{\boldsymbol{N}}^*,\bar{\boldsymbol{A}}^*)$ and initialize $\bar{\boldsymbol{G}}^* = \boldsymbol{G}^*$.

We continue by defining an energy function that reflects

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

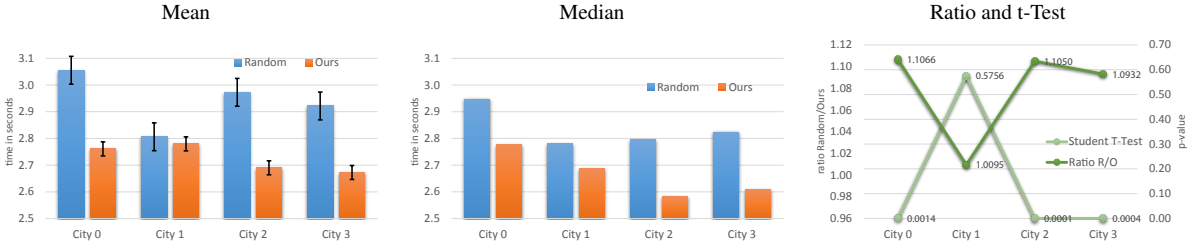


Figure 4: *The results of the response of over 50 users in the layout readability test described in Section 7.1. The black bars in the first chart indicate the standard error. The last chart shows the ratio R/O and the p-value of the Student's t-test.*

the quality of our residual graph as

$$E = \lambda \sum_{(P_i,P_j)} d_{ij} + (1-\lambda)\,\|\bar{A}^*\| \tag{8}$$

where d_{ij} denotes the geographic distance along the shortest path between P_i and P_j and $\|\bar{A}^*\|$ is the number of sections in the residual graph $\bar{G}^*$.

Next we loop over all sections in $\bar{A}^*$, and for each we remove it temporarily from the graph $\bar{G}^*$, which results in a simplified graph $\bar{G}^{*\prime}$. In this step we omit all sections that would split the graph into disconnected components. Having a candidate $\bar{G}^{*\prime}$ for the simplified graph, we again determine all shortest paths between all pairs of POIs in $\bar{G}^{*\prime}$ and again compute the current energy using Equation 8. We add the section back again and proceed with the next one until we have repeated the procedure for all sections. Finally, we choose the simplified graph $\bar{G}^{*\prime}$ that corresponds to the lowest energy $E_{\min}$ and set $\bar{G}^* = \bar{G}^{*\prime}$. Please note that this procedure usually increases, depending on the parameter λ, the overall energy E over the graph compared to the initial full G^*, since by removing a segment we interrupt at least one of the shortest paths. This is true in general, except when there exist more than one shortest route with the same distance, which does not influence the method at all.

In order to further reduce the graph, we repeat the whole procedure on $\bar{G}^*$ until a certain percentage of the original energy (e.g. 105%) is reached, or if none of the edges can be removed anymore without splitting the graph into disconnected pieces. In the latter case we have converged to a spanning tree, which consists of a subset of nodes and edges of G^*. Figure 3 shows the graph simplification results on different stages.

7. Evaluation

To evaluate our method, we have carried out two user studies, one to test the quality of the layout, and one to test the quality of the graph. Furthermore, we provide a quantitative evaluation of the simplified routing graph.

7.1. Layout Readability Test

To provide evidence that our layout is friendly for human perception as postulated by principles **(P2)** and **(P3)**, we performed a user study. We asked 50 participants in an online test to locate a pin on the map given a randomly selected lens (indicated by a highlighting rectangle). We used four test sets from four cities with 12 lenses each. For each of the four cities, we prepared two layouts, one with our method, referred to as O-set, and one with randomly placed lenses, referred to as R-set. Table 1 shows the factor of how much longer the sum of all corresponding distances on the R-set compared to the O-set are (measured in pixels on the map).

For those 8 samples, we asked each user to find a correspondence by clicking on the pin on the map. The order in which the cites and the lenses were presented to the participant was chosen randomly. We checked 11 lenses per sample, which results in 88 altogether. We discarded the first 8 hits from each user, and used the mean of the remaining 80 for statistical evaluation (cf. Figure 4).

We confirmed the normal distribution of the data with a goodness-of-fit test (K-S test). To test the statistical significance of the difference between the O-set (test set) and the R-set (control set), we performed the two-tailed paired Student's t-test. For City0, City2, and City3 the difference was highly significant ($p \leq 0.0014$, $df = 49$). In the case of City1, our results are not significantly better ($p \leq 0.57$).

	City 0	City 1	City 2	City 3
R/O	1.75	1.74	1.66	1.53

Table 1: *Ratio of lens-pin distances on the R-sets compared to O-sets, measured in pixels on the maps.*

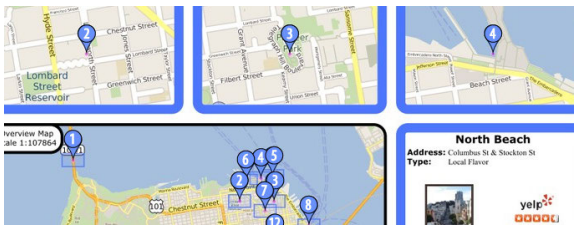


Figure 5: *A distribution of pins and lenses that violates principle **(P3)**.*

Our interpretation of this result is that the lens layout in the O-set for City1 in fact violates the design principles. Generally, principle **(P1)** is violated in all sets, since we test single lenses only. But also principle **(P3)** appears violated,

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

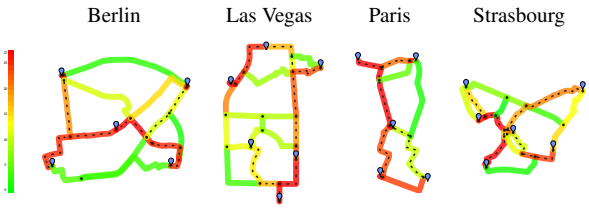


Figure 6: *Evaluation of the user study 2. The color indicates how often a particular street-segment has been selected by the user. The dashed-path is the one determined by our graph-simplification algorithm (cf. Section 7.2).*

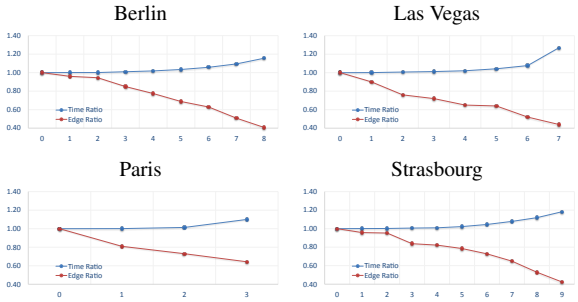


Figure 7: *Charts showing the ratio between the travel time (blue) and the number of edges in the graph (red). Refer to Section 7.3.*

since the lenses (2), (3), and (4) do not mirror the arrangement of the pints on the map, as shown in Figure 5 above (cf. additional material for full test images).

7.2. Graph Reliability Test

We performed another user study to evaluate the quality of our algorithm for the graph simplification. For this we created four maps, each showing a region of a different city. We inserted 5 to 6 POIs and the graph $\boldsymbol{G}^*$ of shortest paths between the POIs into each map. With these maps, we asked 27 persons to fulfill the following task:

> "Connect the points along the purple streets, such that you get the fastest connection from any point to any other point. Use as small number of roads as possible. Note that these requirements are contradictory, and the task is a balancing task. The order is arbitrary."

We counted how often each street segment was selected by the participants and used color coding to indicate streets which were rarely selected (green color) and streets which were selected often (red color). Additionally, we utilized our approach in order to calculate a reference solution. In Figure 6, the results of overlaying our simplified graph with the colored graph is shown. The majority of street segments that were selected by the users are also part of our simplified graph. This is an indication that our algorithm can automatically provide a street graph that consists of street segments which would also be chosen intuitively by real persons. Note that the examples were very small, and it would be far more difficult for a real person to perform the same task for a bigger city region.

7.3. Quantitative Graph Evaluation

We have also tested the performance of our simplified routing graph quantitatively. First, we compute the ratio r_G of the average travel time from A to B for all POIs on the full graph $\boldsymbol{G}^*$ to the average travel time from A to B for all POIs on the simplified graph $\bar{\boldsymbol{G}}^*$ (cf. Section 6). Next, we compare the ratio r_G to the ratio r_S of total number of segments in the respective graphs. Table 2 and Figure 7 show those relations. As can be seen, the average travel time is only 15.59% longer than on the optimal graph in the case of the city of Berlin, on the other hand, the number of segments is reduced to 40.7% of the full graph in the best case. This considerably increases the simplicity and readability of the corresponding graph.

City	Berlin	Las Vegas	Paris	Strasbourg
r_G	1.15585	1.26889	1.09981	1.18273
r_S	0.40704	0.44000	0.64286	0.42408

Table 2: *Ratio r_G of average travel times in $\boldsymbol{G}^*$ to the average travel time in our simplified graph $\bar{\boldsymbol{G}}^*$ and ratio r_S of total number of segments in the respective graphs.*

8. Results and Discussion

Figure 9 shows examples of our brochures. Further results are added as supplemental material. For all the results, the input is a set of points of interest that have been delivered automatically by a travel-related web-service (i.e., www.yelp.com) to a certain combination of keywords (e.g., "museum"), and filtered by the user.

System Parameters. Table 3 lists all parameters of our system. Values denoted as 'Default' were fixed for all our experiments and examples.

Parameter	Symbol	Examples	Set By
POIs	$\boldsymbol{P}$	-	User
Format	S	A3, A4	User
Resolution	G	9x5	User
Map	$M_0, \ldots, M_n$	-	User
DPI	-	300	Default
Map Scales	s_{M_0}, s_{M_1}, s_D	14, 15	Default
Variables	$\lambda, \lambda_1, \lambda_2$	0.5, 0.1	Default

Table 3: *Parameters in our system.*

Comparison to Human Designers. We have reviewed several travel brochures created by human designers. Our general observations are:

- There are many tourist brochures where POIs are marked with numbers and more details about the POIs are listed on the side of the map. The details can be a longer description, photos, or just the name.
- Multiple tourist brochures use insets of maps of different resolutions, but existing maps do not use as many as we do. Detail maps for single POIs were not observed.
- Several tourist brochures are based on an artistic abstraction of a map and some use traditional maps like we do.

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

City	$\|Seg_{in}\|$	$\|Seg_{out}\|$	t_{simpl} [s]
Seattle	400	46	8.927
Los Angeles	297	44	5.202

Table 4: *Performance of the graph simplification (cf. Fig. 9).*

City	$\|\boldsymbol{P}\|$	#BIP Calls	t_{pl} [s]
Seattle	30	7	0.232
Los Angeles	30	19	0.872

Table 5: *Performance of the layout computation (cf. Fig. 9). The #BIP calls refers to the number of how often the k-means (cf. Sec. 4.4) algorithm, and thus also the BIP, have been run.*

- Several tourist brochures try to place details about the POIs directly on the map.
- Some tourist brochures color map markers differently depending on their category (e.g. sightseeing, food, shop).

Figure 8 shows a tourist brochure of Brugan, Belgium, which was designed by a human. This brochure uses color coding for the categories of the POIs. The numbering of the markers in the map, however, is not related to the position of the corresponding detail information on the side of the map. This makes it quite hard to find a marker for a given information block. In contrast, our system tries to place the detail information in form of the lenses in the grid according to the position of the markers in the overview map. Additionally, our lenses are enumerated in a top-to-bottom and left-to-right fashion in order to make them easier to locate.

Another issue is the handling of spatial density of the contained information. Our system tries to encapsulate the detail of a region with a high number of POIs into one multi-POI lens. This has two benefits: it reduces visual clutter, and bundles local information spatially. In the example below, however, the city center is populated quite densely with numerous markers, but it does not provide a higher scale map, making it more difficult to read. Moreover, the corresponding detail blocks are scattered around. For this reason the user needs to establish several cross-references to get the detail information of the POIs in one particular region.

Figure 8: *A hand-made brochure from www.use-it.be.*

Limitations. Indeed, our system can handle most compositions of POIs. Nonetheless, in some cases it can happen that the generated layouts do not obey the design principles, as is the case in Section 7.1, where the layout of City1 seems not to be perceived as better as the random layout. Our system can estimate layouts by calculating the corresponding energy value. However, a low energy value might not always guarantee that a map is perceived as more intuitive by humans. Changing the value of λ in Eq. 1 might provide a remedy, but we do not expose the parameter to the user.

Another problem arises during routing by the occurrence of one-way streets. If the graph consisting of all shortest paths is minimized subject to the one-way streets, it is not always possible to reach a spanning tree since it is likely that a POI can not be left using the same street that was used to reach it. For convenience, we did not further address this problem and regarded all streets as bi-directional. This is not basically wrong, since one-way streets can usually be used by pedestrians or even cyclists in both directions.

Lens Discrimination. We have experimented with different effects to provide best visual discrimination of the lenses. Our first draft solved the four-color problem such that neighboring lenses were always of different background color. In fact, this turned out to be perceived as too visually cluttered in preliminary user experiments. Thus, we decided to work with a distance between lenses which is of the same width as the colored border of each lens.

Interactive GUI. We have implemented a prototypic GUI where the user can enter the location and search terms in order to receive a set of POIs from a web-database. Moreover, she can resize and place the map, then the grouping and the layout of the lenses is computed automatically.

Implementation and Performance. We have implemented a prototype of the brochure generator in C++ using MS Visual Studio 2010. We also used OpenStreetMap as the GIS database and the Mapnik library for the vector-graphics map rendering purposes. For efficient graph computations, we used the Lemon Graph Library. For the GUI we used C#/WPF. Table 5 shows the running time of the layout procedure (Sec. 5), and Table 4 shows the running time of the graph simplification algorithm (Sec. 6).

9. Conclusions

We have presented a system for the automatic generation of tourist brochures with route information, which provide information about a set of points of interest in a particular region in a focus-and-context style by the use of detail lenses.

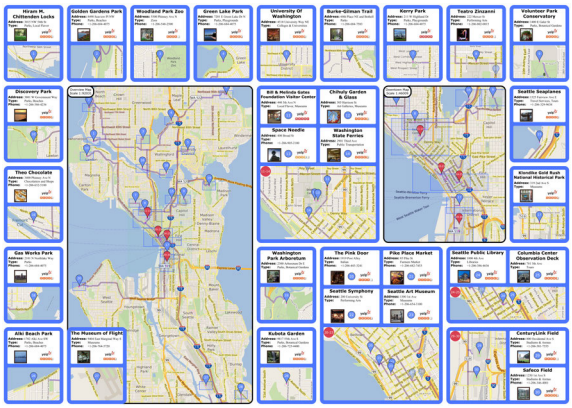
In addition, we contribute an algorithm for route-aware simplification of routing graphs that provide near-optimal routing directions on the one hand, and are simple and clutter-free on the other. We showed that such graphs match well with those selected by humans for traveling between arbitrarily distributed POIs. Moreover, we introduce a novel layout algorithm that optimizes a layout of a number of elements of different sizes in a discrete grid under specific neighborhood constraints. In the future we plan to extend the implementation of our system to real-time interactive route planning on desktop and mobile devices.

Acknowledgments

We thank Matthias Bernhard and Michael Hecher for discussions, all participants of the user studies for time and pa-

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

Brochure of Seattle



Brochure of Los Angeles

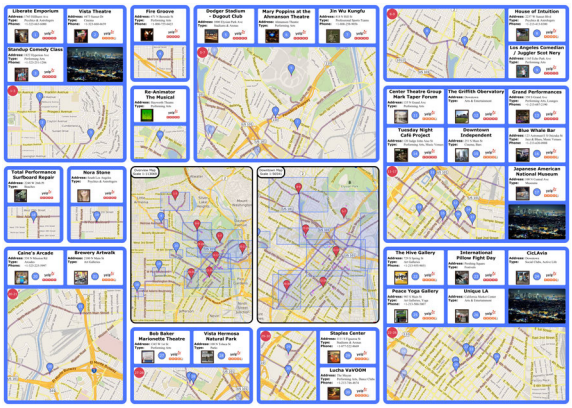


Figure 9: *Examples of our tourist brochure for Seattle and L.A. For more results please refer to additional material.*

tience, and the anonymous reviewers for valuable comments. This work was funded by the FWF, project no. P23237-N23.

References

[ALB11] AGRAWALA M., LI W., BERTHOUZOZ F.: Design principles for visual communication. *Communications of the ACM 54*, 4 (Apr. 2011), 60. 2

[AS01] AGRAWALA M., STOLTE C.: Rendering effective route maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01* (New York, New York, USA, Aug. 2001), ACM Press, pp. 241–249. 2

[BFSS07] BAST H., FUNKE S., SANDERS P., SCHULTES D.: Fast routing in road networks with transit nodes. *Science (New York, N.Y.) 316*, 5824 (Apr. 2007), 566. 2

[CBGS09] CHEN W.-C., BATTESTINI A., GELFAND N., SETLUR V.: Visual summaries of popular landmarks from community photo collections. In *Proceedings of the seventeen ACM international conference on Multimedia - MM '09* (New York, New York, USA, Oct. 2009), ACM Press, p. 789. 2

[CBHK09] CRANDALL D. J., BACKSTROM L., HUTTENLOCHER D., KLEINBERG J.: Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web - WWW '09* (New York, New York, USA, Apr. 2009), ACM Press, p. 761. 2

[DFV*08] DI IORIO A., FURINI L., VITALI F., LUMLEY J., WILEY T.: Higher-level layout through topological abstraction. In *Proceeding of the eighth ACM symposium on Document engineering - DocEng '08* (New York, New York, USA, Sept. 2008), ACM Press, p. 90. 2

[FM98] FODNESS D., MURRAY B.: A Typology of Tourist Information Search Strategies. *Journal of Travel Research 37*, 2 (Nov. 1998), 108–119. 2

[GASP08] GRABLER F., AGRAWALA M., SUMNER R. W., PAULY M.: Automatic generation of tourist maps. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 2

[Gur13] GUROBI OPTIMIZATION, INC.: Gurobi optimizer reference manual, 2013. 6

[JLS*03] JACOBS C., LI W., SCHRIER E., BARGERON D., SALESIN D.: Adaptive grid-based document layout. In *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03* (New York, New York, USA, July 2003), vol. 22, ACM Press, p. 838. 2

[KAB*10] KOPF J., AGRAWALA M., BARGERON D., SALESIN D., COHEN M.: Automatic generation of destination maps. *ACM Transactions on Graphics 29*, 6 (Dec. 2010), 1. 2

[KCJ*10] KARNICK P., CLINE D., JESCHKE S., RAZDAN A., WONKA P.: Route visualization using detail lenses. *IEEE transactions on visualization and computer graphics 16*, 2 (Jan. 2010), 235–47. 2

[LFC04] LUO M., FENG R., CAI L. A.: Information Search Behavior and Tourist Characteristics. *Journal of Travel & Tourism Marketing 17*, 2-3 (Feb. 2004), 15–25. 2

[LFN04] LOK S., FEINER S., NGAI G.: Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interface - IUI '04* (New York, New York, USA, Jan. 2004), ACM Press, p. 101. 2

[MSL*11] MERRELL P., SCHKUFZA E., LI Z., AGRAWALA M., KOLTUN V.: Interactive furniture layout using interior design guidelines. *ACM Transactions on Graphics 30*, 4 (July 2011), 1. 2

[PF06] PAN B., FESENMAIER D. R.: Online Information Search. *Annals of Tourism Research 33*, 3 (July 2006), 809–832. 2

[SMSW90] SNEPENGER D., MEGED K., SNELLING M., WORRALL K.: Information Search Strategies By Destination-Naive Tourists. *Journal of Travel Research 29*, 1 (July 1990), 13–16. 2

[SS07] SANDERS P., SCHULTES D.: Engineering fast route planning algorithms. In *WEA'07 Proceedings of the 6th international conference on Experimental algorithms* (Rome, Italy, June 2007), Springer-Verlag, pp. 23–36. 2

[SSN*05] SHIRAISHI T., SHIBATA M. N. N., NAGATA M., SHIBATA N., MURATA Y., YASUMOTO K., ITO M.: A personal navigation system with a schedule planning facility based on multiobjective criteria. In *Proc. of 2nd Int. Conference on Mobile Computing and Ubiquitous Networking* (2005). 2

[YYW*12] YEH Y.-T., YANG L., WATSON M., GOODMAN N. D., HANRAHAN P.: Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Transactions on Graphics 31*, 4 (July 2012), 1–11. 2

[Zal96] ZALATAN A.: The determinants of planning time in vacation travel. *Tourism Management 17*, 2 (Mar. 1996), 123–131. 2

[ZYZ*13] ZHENG Y.-T., YAN S., ZHA Z.-J., LI Y., ZHOU X., CHUA T.-S., JAIN R.: GPSView: A scenic driving route planner. *ACM Transactions on Multimedia Computing, Communications, and Applications 9*, 1 (Feb. 2013), 1–18. 2

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.