# Automatic generation of tourist maps

**Master Thesis**

**Author(s):**
Grabler, Floraine

**Publication date:**
2008

**Permanent link:**
https://doi.org/10.3929/ethz-a-005632669

# Automatic Generation of Tourist Maps

Floraine Grabler

*Master Thesis*

Advisors: M. Agrawala, B. Sumner and M. Pauly

Department of Computer Science
UC Berkeley, ETH Zurich

April 1, 2008

# Abstract

Tourist maps are essential resources for visitors to an unfamiliar city because they visually highlight landmarks and other points of interest. Yet, hand-designed maps are static representations that cannot adapt to the needs and tastes of the individual tourist. In this thesis we present an automated system for designing tourist maps that selects and highlights the information that is most important to tourists. Our system determines the salience of map elements using bottom-up vision-based image analysis and top-down web-based information extraction techniques. It then generates a map that emphasizes the most important elements, using a combination of multiperspective rendering to increase visibility of streets and landmarks, and cartographic generalization techniques such as simplification, deformation, typification and displacement to emphasize landmarks and de-emphasize less important buildings. We show a number of automatically generated tourist maps of San Francisco and compare them to existing automated and manual approaches.

# Acknowledgments

I would like to thank the people who supported me during my work on this thesis. I am particularly grateful to Maneesh Agrawala, my advisor at U.C. Berkeley. His experience and expertise in visualization are invaluable and his encouragement and support were crucial to the successful completion of this thesis. I would like to thank Maneesh for introducing me to research and encouraging me to pursue my own ideas. I look forward to continue working with him and pursue a Ph.D. at Berkeley.

Many thanks to my ETH-advisor Mark Pauly and Bob Sumner for their helpful suggestions and numerous fruitful discussions. Mark was also the first person to teach me computer graphics. His encouragement and enthusiasm for graphics pushed me to pursue in this direction. Bob was always a great person to work with, not only for his helpful advice but also for his great sense of humour.

Special thanks to Microsoft for providing us with the necessary data.

Last but not least, I would like to thank the vendor from the corner store and the janitor of Soda Hall who often appeared more concerned about my deadline than I was and occasionally offered me late night snacks.

I would also like to thank all the professors that have helped me along the way in my Bachelor and Master studies at ETH Zurich. I am deeply grateful to Hans Hinterberger who has always been very supportive and encouraging since my third semester at ETH. He valued my enthusiasm for school and pushed me to apply as a visiting scholar to Berkeley. Organizing my visit to Berkeley and extending my stay has been a true challenge. Especially Hans Hinterberger and Markus Gross, but also Mark Pauly and Walter Gander have made it possible. I am very grateful to Markus Gross for his invaluable guidance and advice on topics related to graduate school. I thank Hans Dubach for having contributed in making my studies challenging until the very end.

ETH and Berkeley have been good places to work because of the many friends I have made. In particular I would like to thank: Hanni Wolf who has been a great comrade through the hardship of undergraduate school. By sharing endless discussions about movies, art and life, joining my sarcasm and cooking delicious Japanese meals, she has kept me smiling. Jason Sanders, my former office mate and coffee brother, for making our lab a great place to spend days in (and before deadlines nights). Rose-Line Werner and Moritz Baecher for the great teamwork in many course projects and their incessant optimism and support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Tourist maps are essential resources for visitors to an unfamiliar city because they visually highlight landmarks and other points of interest such as museums, restaurants, parks, and shopping districts. The most effective tourist maps are carefully designed to present this information so that visitors can easily navigate to the places they are most interested in. Yet, designing and rendering such tourist maps is a time-consuming process that requires expert map design skills.

Moreover, hand-designed maps are static representations that cannot adapt to the needs and tastes of the individual tourist. Points of interest can differ significantly from person to person. While visually distinctive buildings and environmental features can serve as general-purpose landmarks, one tourist may be most interested in shopping, while another may want to see nearby restaurants. In designing a map, the first challenge for mapmakers is to determine the importance of these elements to the tourists that will use the map.

Tourists often use a combination of public transportation and walking to move from one place to another. Therefore, an effective tourist map must provide rich visual representations of landmarks and points of interest to help tourists quickly identify where they are located and determine the best (most interesting) route to their destination. For example, mapmakers use multi-perspective rendering to increase the visibility of streets and landmarks. Similarly, they use a variety of cartographic generalization techniques to increase the recognizability of landmarks and emphasize the most important elements in the map. Thus, the second challenge for mapmakers is to choose a set of rendering and cartographic generalization techniques that emphasize the most important landmarks and points of interest while de-emphasizing or eliminating irrelevant elements.

In the last decade, digital maps such as those provided by Microsoft Live (www.live.com), Google Maps (maps.google.com) and Yahoo Maps (maps.yahoo.com) have become increasingly popular. Cities are usually in a perpetual state of construction, renovation and renewal in which streets and buildings are created, destroyed and reshaped. One advantage of such digital maps over hand-designed maps is that they are based on continually updated geometric models of the streets and buildings in the city and therefore usually reflect the most up-to-date information. However, these digital maps do not adequately address the two primary challenges of creating tourist maps and suffer from several problems, including:

***Too much clutter.*** Digital maps do not identify the important landmarks and points of interest. Instead they display every street and building in the region, making it difficult for tourists to filter out the information they need and localize the important features of a city.

***Lack of organization/legibility.*** Digital maps fail to organize the map into distinct and easily recognizable units, such as landmarks, streets, blocks, and parks. As a consequence the layout of the city is complex to understand and tourists experience difficulties in wayfinding tasks.

***Poor visibility.*** Digital maps use a single perspective that makes it difficult to see both the street network and the buildings in a single view. In contrast, mapmakers often use multiple perspectives so that both the streets and the (visually salient) street-side facades of buildings are clearly visible.

***Poor generalization.*** Digital maps do not generalize and highlight the important landmarks of the city. All buildings are presented at the same level of detail, making it difficult to distinguish the most important landmarks.

In this thesis we present an automated system for generating tourist maps that addresses the two primary mapmaking challenges highlighted earlier, and significantly reduces the problems inherent in current digital maps (see Figure 1.1). The input to our system consists of a geometric model of a city, including streets, bodies of water, parks and buildings (with textures). Users can optionally specify categories of places (i.e. restaurants, shopping, ...) they are interested in. Our system automatically determines the salience and importance of map elements using bottom-up vision-based image analysis and top-down web-based information extraction techniques. It then generates a map that emphasizes the most important elements, using a combination of multiperspective rendering to increase visibility of streets and landmarks, and cartographic generalization techniques such as simplification, typification, deformation, and displacement to emphasize landmarks and de-emphasize less important buildings.


***Figure 1.1:*** *(top) A digital map of San Francisco from Microsoft Live (www.live.com) suffers from clutter, lack of organization and legibility, poor visibility of roads and buildings and poor highlighting of important landmarks. (bottom) A tourist map generated automatically by our system emphasizes the information required by tourists.*

# Chapter 2

# Tourist Map Design

In order to design more effective tourist maps we follow the approach of Agrawala and Stolte [1] and begin by analyzing prior work on mental representations of cities [19, 30] as well as collections of the best hand-designed maps [31, 37, 12] and cartographic text books [17]. From this analysis we extract the importance of the elements in a tourist map and identify a set of principles for rendering useful tourist maps.

## 2.1 Important information

In his classic book, *The Image of the City*, Kevin Lynch [19] identifies five elements that people use to form mental representations of cities: *landmarks, paths, districts, nodes* and *edges*. Indeed, these elements have a number of properties that make them essential in navigational tasks and in the general understanding of a new environment. Figure 2.1 shows an example of each element.

***Landmarks.*** Large physical objects often act as reference points in the environment. While our work focuses on buildings, other objects such as bridges and mountains can also serve as such landmarks. The principal characteristic of landmarks is that they are uniquely memorable in the context of the surrounding environment. Sorrows and Hirtle [30] consider three subcategories of landmarks:

- *Cognitive landmarks* are semantically meaningful because of either their cultural or personal significance. For example, the apartment of a famous author is culturally significant, while a particular restaurant may be personally significant.
- *Visual landmarks* are buildings that are distinctive because of their visual characteristics, such as color or shape.
- *Structural landmarks* are memorable because of their spatial location. Examples include buildings located at decision points such as street intersections, bus stations or around town squares.

All three subcategories indicate important features of well designed tourist maps. While cognitive landmarks characterize a place or reflect the personal interests of a viewer, visual and structural landmarks are essential for navigation. Such landmarks provide additional context for navigating through unfamiliar areas and enable users to localize themselves within the surrounding environment [5, 23].

***Paths.*** Roads, walkways, transit lines, canals, railroads, etc. are paths through the environment on which people may freely assemble, interact, and move about. In this work we focus on roads, since they are the predominant paths for urban navigation.

***Districts.*** Some areas of a city may have common identifying characteristics along a variety of dimensions including building type, building use, types of inhabitants etc. Neighborhoods such as Chinatown, Little Italy and the Mission are examples of such districts.

***Nodes.*** In many cities, parks, town squares, beaches and busy intersections are points where people tend to congregate. Such nodes are particularly relevant for tourists because they are good places to mingle with the local population.

***Edges.*** Many cities contain elements such as rivers, city walls, and roads that serve as linear breaks in continuity between regions. Such edges can be barriers which close one region off from another, or seams along which two regions are related and joined together.

Edges are often defined by the boundaries of the other four city elements. For example a highway serves as a barrier but also as an important path. Our system does not explicitly treat edges, but handles them implicitly by working with the other four elements.

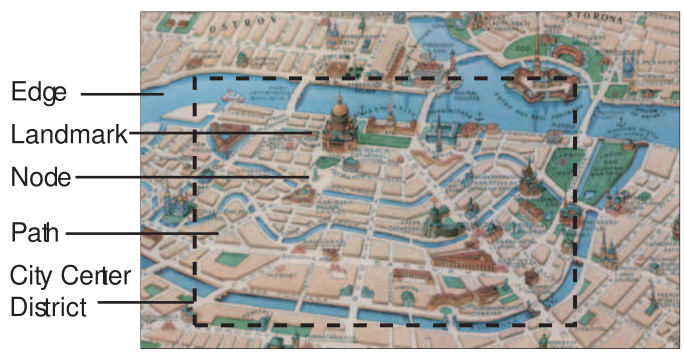



***Figure 2.1:*** *Examples of the five city elements: landmarks, paths, districts, nodes and edges in a map of St-Petersburg.*

## 2.2 Rendering

### Multiperspective Maps

Cartographers often use multiple viewpoints and perspectives in order to increase the visibility of important elements of the map. A common approach is to combine an orthographic, top-down, plan view for the streets and ground plane, with either an oblique or perspective projection for the buildings. The top-down view of the ground plane eliminates foreshortening distortions and thereby allows users to better understand the layout of the road network. Cartographers choose oblique or perspective views for the landmarks to ensure that the street-side facades of the buildings are clearly visible. Figure 2.2 shows an example of a multiperspective map combining an orthographic view of the ground plane and perspective and oblique views of the landmarks.

Oblique and perspective projections have complementary advantages. With an oblique projection the size and shape of the building does not depend on the distance to the viewpoint. Agrawala et al. [2] point out that setting the oblique projection image plane parallel to the ground plane ensures that the building footprints retain their true shape and thereby facilitate some size and area comparisons. However, the lack of perspective cues can also make the buildings appear unrealistic. Using a perspective projection for the buildings prevents this problem, but makes it difficult to visually compare size and area of buildings.

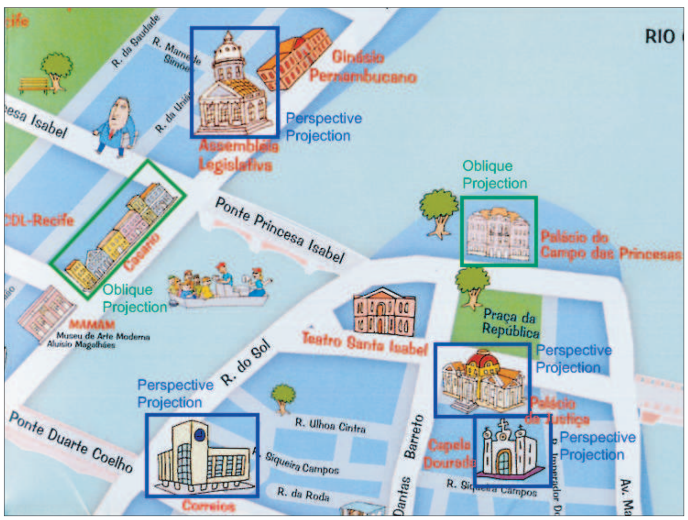



*Figure 2.2: An example of a multiperspective map that uses an orthographic projection for the ground plane and a perspective (blue boxes) and oblique projection (green boxes) for the landmarks.*

#### Generalization

Mapmakers use a variety of cartographic generalization techniques including simplification, typification, displacement, deformation and selection to improve the clarity of the map and to emphasize the most important information, while preserving spatial relationships between map objects [25, 20, 7]. In this work, we consider the generalization techniques mapmakers apply to buildings in hand-drawn tourist maps and show how these techniques improve map usability.

***Simplification.*** Mapmakers use simplification to de-emphasize less important buildings or to reduce the complexity of scaled down buildings and thus avoid artifacts.

***Displacement.*** Artists often widen roads in order to emphasize the paths in an unknown area. As a consequence, the space available for buildings in the block is limited. A common technique is to displace the buildings in order to avoid overlaps with neighboring streets or buildings.

***Deformation.*** Mapmakers often increase the size of important landmarks while decreasing the size of less relevant buildings.

***Removal.*** Handdrawn maps often remove less relevant buildings to reduce the complexity of maps and to gain space to emphasize more important buildings.

***Typification.*** Artists often typify buildings by recognizing similar buildings and giving them the same appearance. Typified buildings are perceived as a group and not as a set of individual buildings with minor differences and thus help a person understand a map by reducing its complexity.

# Chapter 3

# System Overview

We have developed an automated system for generating tourist maps based on the design principles presented in Section 2. As shown in Figure 3.1, the input data to our system consists of a complete geometric model of a city in lat/lon coordinates, roads categorized by type into street, major roads, arterial, ramp and highway. A traffic map separates the streets into four discrete levels of traffic. A ground plane image segments the city into waterways, parks and ground. Finally, as explained in Section 4.1, our system parses a set of webpages[1] from which we extract semantic information about buildings including their category (museum, restaurant etc.), neighborhood, and a user ranking for each building.

From this data our system precomputes building footprints and block boundaries, where a block refers to the smallest area delimited by roads. We determine the building footprints by projecting the building triangles on the ground plane and tracing out the outline of the building in pixel space. We compute the block boundaries by triangulating the road network and applying a flood-fill over the triangles stopped by the street segments.

After preprocessing we identify landmarks, paths, districts, and nodes, as well as importances for each of these elements (Section 4). To generate a tourist map, a user specifies a city, or a region within a city, and the landmark categories of personal interest (i.e. restaurants, shopping, etc.). The system then designs and renders a tourist map that covers the specified region and highlights the personal points of interest (Section 5).

[1] www.openlist.com, www.tripadvisor.com, and travel.yahoo.com

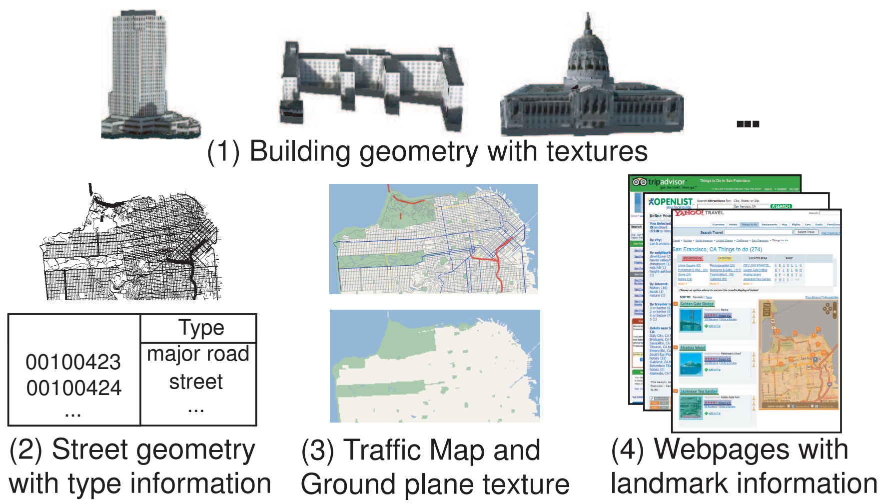



***Figure 3.1:*** *The input data to our system.*

# Chapter 4

# Computing Importance

Our system automatically extracts landmarks, paths, districts and nodes and their relative importance by constructing a set of features from the underlying city data. Following the approach of Sorrows and Hirtle [30], we divide the features into three categories; *semantic, visual* and *structural*. The semantic features are computed via web-based information extraction techniques, while the visual and structural features are based on low-level analysis of the city geometry, textures, and ground plane image. Every feature is associated with a score, where a high score indicates a high importance.

We compute the overall importance for each map element as a weighted sum of the feature scores. While users must pick the landmark categories they are most interested in, they can also manually set the weight for each feature in the weighted sum. Alternatively we provide a set of experimentally chosen default weights.

Several other papers [27, 16] proposed a similar approach to formally specify the landmark saliency based on a set of features. Our approach differs from these works by the set of features we use, by making the semantic landmarks dependant on the user as well as how we combine the features. We also use a more global approach where we extend the classification into semantic, visual and structural features to other city elements and study how the importance of each element interacts with other elements. For example, the importance of a landmark depends on the importance of the roads it is facing.

Similar to our work, Zipf [42] studied how tourist maps can adapt to user preferences. In particular he looked at how the culture, location and interests of the user change a map on a mobile device. Unlike his approach we aim at generating 2D static maps and therefore do not take into account the current location of a user. Another difference is that in his system the elements that are rendered given the user's interests are predefined in an XML-file and not determined automatically by modifying the weights of the semantic features.

## 4.1 Landmarks

We base our landmark features on Appleyard [3] who conducted a survey asking people what buildings they remembered along a road. From this survey he suggested a set of salient landmark features that we incorporate into our system.

#### Semantic Features

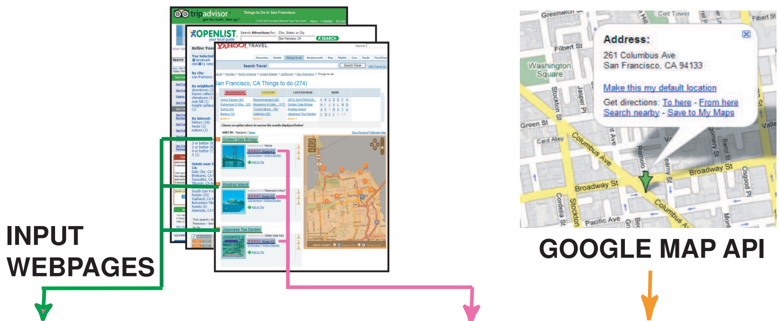
Identifying semantically meaningful features requires human knowledge of the cultural or personal significance of the landmark. While such semantic meaning cannot be determined from the geometric city data alone, the Internet provides a vast collection of semantically structured data from which we extract the appropriate features. A number of recent techniques exploit the web to extract such semantics in the context of photo manipulation [29, 11]. Similarly Tezuka and Tanaka [33] apply text-mining techniques to extract landmark information from unstructured web documents.

We extend this idea to parse landmark information from three travel-related websites. As shown in Figure 4.1 we extract the following attributes: name, lat/lon position, address, district, category (tourist attraction, museum, business facility, shopping, restaurants etc.) and user ranking (0 to 5 stars). When the travel website does not contain the landmark position our system queries the Google Map interface with the building address to retrieve the position. We use the lat/lon position to associate each landmark with the corresponding geometry from our city data set. Finally, we take the average user ranking of the landmark across the three websites as its semantic score. Figure 4.4(top) shows the most important semantic landmarks in an area.

#### Visual Features

Buildings with distinctive visual features (color, shape, etc.) that differ significantly from the appearance of other buildings in the local area are likely to be remembered as landmarks. We distinguish three types of visual characteristics that strongly influence the saliency of a building: *facade color, shape complexity* and *building height*. We first define a set of features that quantify the presence of these characteristics in a building and then explain how we associate scores with these features.

***Facade color.*** Some buildings are memorable because their facades are colored differently from the surrounding buildings. As shown in Figure 4.2, to extract the wall color of a building, we first separate building geometry into roof and wall triangles. A building triangle $i$ with normal $n_i$ is labeled as a wall, if $\mathbf{n}_i \cdot (0,0,1) < 0.5$, and as a roof otherwise. Next, for all wall triangles we apply Felzenszwalb's [9] color-based segmentation algorithm on the corresponding texture map. We approximate the color of each segment by its average color to smooth out small color irregularities and noise in the texture. Finally, to avoid dark shadows in the texture, we set the facade color to the the brightest color in the segmented texture, where we consider only the segments that




| Name | Category | District | Rating | Address | Position |
|---|---|---|---|---|---|
| City Lights Booksellers | Business Facility | North Beach | 3 | 261 Columbus | lat: 37.7975 lon: -122.4062 |
| Japanese Tea Garden | Tourist Attraction | Golden Gate Park | 4.5 | 7 Tea Garden Dr | lat: 37.7692 lon: -122.4698 |

SUMMARY DATABASE

***Figure 4.1:*** *We extract landmark attributes from three input webpages. When the webpages only provide the address of a landmark we look up the exact lat/lon position by querying the Google Map API.*

have an area larger than 5% of the total wall area. The color value is the L component of the facade color in the LAB color space.

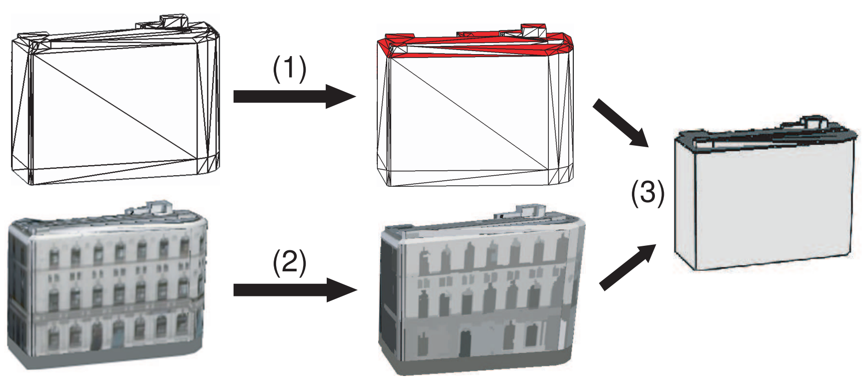



***Figure 4.2:*** *Extraction of building color. (1) We first distinguish between roof and wall triangles, (2) segment the texture and average the color of every segment, (3) and then we set the wall color to be the color of the brightest component wall area.*

***Shape complexity.*** Many buildings have a simple, regular, rectangular shape. The more a building deviates from this standard shape, the more salient it becomes as a landmark. We measure the shape complexity using two features. The first feature measures the rectangularity of a building as the ratio of the volume of the building's bounding box to the true volume of the building [24]. The second feature measures the angle variation of the building triangles and is computed as the sum of absolute dihedral angles weighted by the length of the triangle edge. Figure 4.3 illustrates the difference between these two shape complexity features.

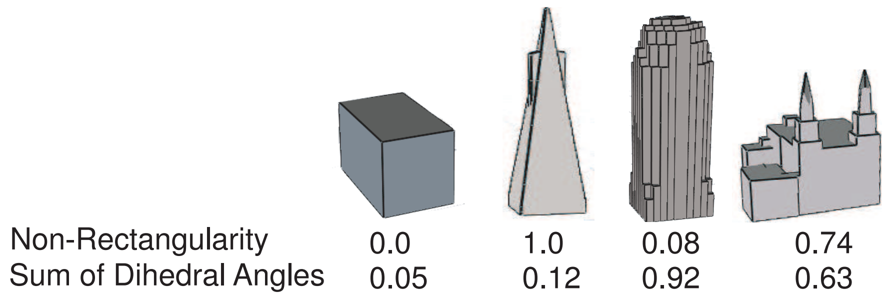



***Figure 4.3:*** *Shape complexity features. The first rectangular building has low complexity by both measures. The second building is unusual because it is highly non-rectangular, but it also has a low sum of dihedral angles. The third building is fairly rectangular but also has many corners and therefore has high shape complexity. The last building is has high shape complexity by both measures.*

***Building height.*** Taller buildings are more likely to be visible from further away. Therefore we create building height as a visual feature with taller building getting higher value.

In the context of visual features, a building only becomes a landmark if it differs significantly from its neighboring buildings. To compute the score of each visual feature, we thus estimate the deviation from the median using the measure proposed by Nothegger et al. [26]:

$$score = \frac{\mid x - med(x) \mid}{med(\mid x - med(x) \mid)},$$

where $x$ is the individual measure and $med(x)$ denotes the median. To compute the median we define the local neighborhood of each building to include all other buildings within 50m. Figure 4.4(middle) shows the most important visual landmarks in an area.

#### Structural Features

Several cognitive psychology studies have shown that travelers are more likely to take note of buildings located at street intersections and around town squares as they learn to navigate a new city [6, 18, 23]. The spatial locations of these buildings makes them especially relevant for navigation and therefore we identify both of these types of structural landmarks.

***Buildings at intersections.*** Buildings at important intersections are more prominent than others. We compute a score for each intersection as the sum of the importance scores of all streets (see Section 4.2 on computing road importance) meeting at that intersection. Thus, we favor intersections of important roads or places where many roads meet. Finally, we transfer the importance of an intersection to the nearby buildings using a Gaussian weighting factor based on the distance between the intersection and the building. For each building, we accumulate influences from all nearby intersections.

***Buildings around squares.*** To identify buildings located around town squares we first extract the squares and their importance as explained in Section 4.4. The importance of the square is then transferred to the buildings nearby analogously to the intersection importance.

Figure 4.4(bottom) shows the most important structural landmarks in an area.

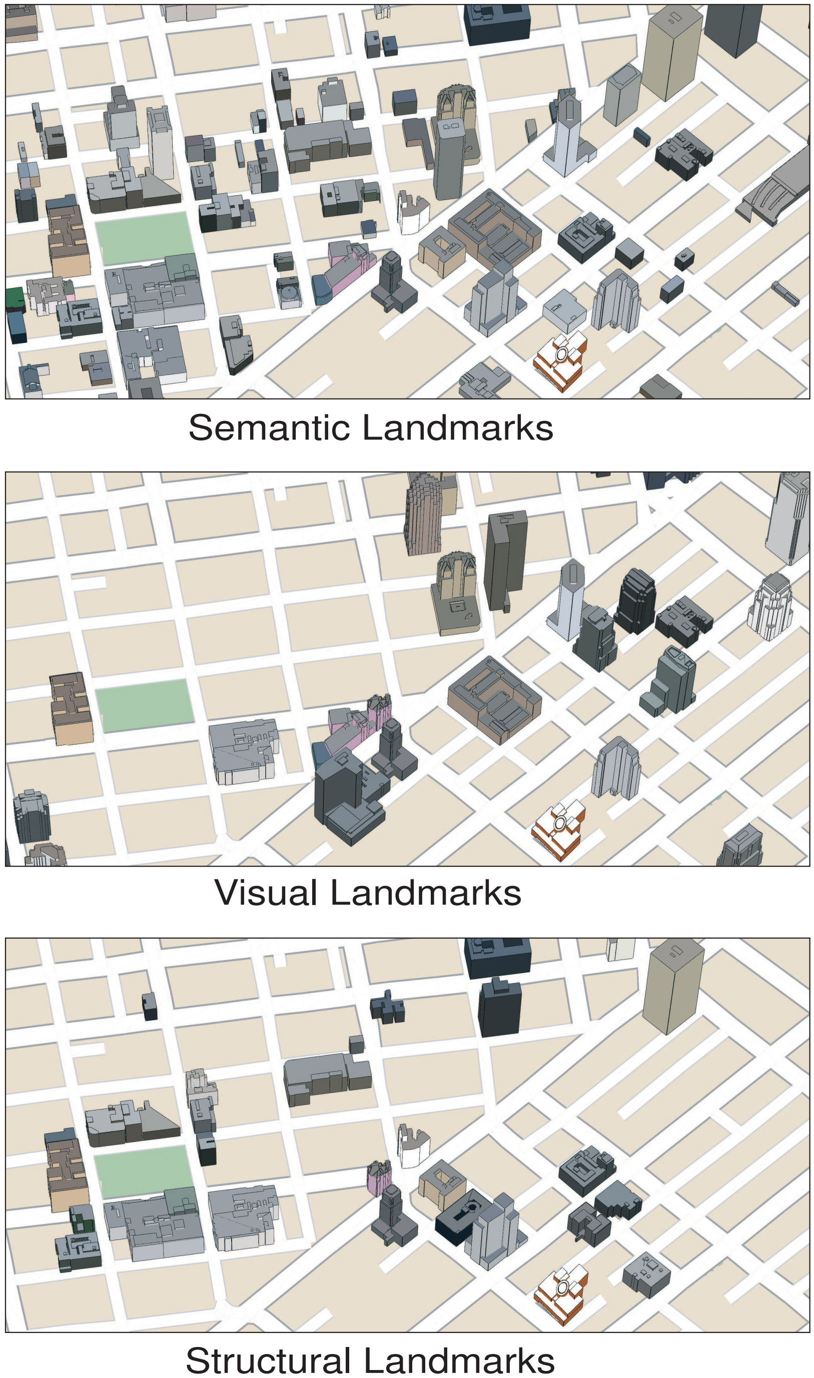



***Figure 4.4:*** *Comparison of semantic, visual and structural landmarks in an area.*

## 4.2 Roads

We use semantic and structural features to identify the important roads. While semantic features are the same for all maps, the structural features depend on the points of interest chosen by the user.

#### Semantic Features

Previous work on the selection of important roads has focused on inferring semantic importance indirectly from the topology and geometry of the roads [21, 35, 36, 14]. In contrast, our system directly accesses this semantic information from the input data or the web.

***Categories.*** Roads are often classified into a discrete set of categories according to their usage. For example, each road in our input data is categorized as either a street, major road, arterial, ramp or highway. The respective scores for each categories are 0.25, 0.5,0.75,1.0,1.0.

***Traffic.*** Customary travel is one of the strongest influences in the importance of a road [19]. Major access lines such as the Bay Bridge or the Presidio Parkway in San Francisco are key features of the mental map. We retrieve the information about road use from a traffic map (travel.yahoo.com) that classifies road segments into four discrete levels of traffic according to their usage during rush hours. The traffic score of a road segment corresponds to its normalized traffic level.

#### Structural Features

We specifically design the structural features to select for roads that facilitate navigation to and around interesting areas for tourists. Our roads are represented as polylines and we compute importance scores for each linear segment of the road. The segment importance indicates how relevant a road is locally and influences the importance of nearby landmarks, as explained in Section 4.1.

***Landmark proximity.*** For tourists, the roads closest to their personal points of interest are especially important because they facilitate navigation. To compute this feature, we first estimate the semantic importance for the landmarks chosen by the user as points of interest (restaurants, shopping, etc.) as described in Section 4.1. We then subsample every road segment at 2m intervals and transfer the semantic importance of nearby landmarks to these road sample locations using a Gaussian weighting factor based on the distance between the road sample and the landmark location. At each road sample point we accumulate the influence of all nearby landmarks. Finally, for each linear segment of a road we compute the average score of its subsampled points. This feature emphasizes the roads immediately surrounding a landmark.

***Roads connecting landmarks.*** Since tourists often want to circulate between different sights of the city, it is helpful to emphasize the connecting roads. We query the Virtual Earth map control API (http://dev.live.com/virtualearth/sdk) to obtain the route between each pair of landmarks. The score for each road segment is then computed as the number of times that segment is part of a landmark to landmark route. Unlike landmark proximity, this feature can increase the importance of roads that do not have many landmarks on them, but are often used to go between interesting parts of the city.

## 4.3 Districts

#### Semantic Feature

In identifying the semantic features for each landmark we also extract the district or neighborhood the landmark belongs to (see Section 4.1). Districts usually do not have a hard or precise boundary and therefore we simply identify the district center, by searching for the location with the highest density of landmarks from that neighborhood. We determine this location by iteratively computing the mean location of all landmark positions in that district and rejecting outliers at each iteration if necessary.

## 4.4 Nodes

#### Semantic Feature

Some nodes such as squares, parks and lakes are also considered landmarks and we extract them by parsing travel websites in exactly the same way we identify semantic landmarks (see Section 4.1). However this extraction process provides only a point location for the node. To compute the 2D extent of squares we expand the point location to the entire surrounding block. Similarly we use a color-based flood-fill on the ground plane texture seeded at the point locations of parks and lakes to determine their 2D extents. The score of a node, as for the semantic landmark feature, corresponds to the average user rating indicated by the input websites.

# Chapter 5

# Rendering

## 5.1 Multiperspective Maps

Multiperspective maps have the advantage of depicting roads and the visually salient facades of buildings in a single view. To create such multiperspective maps we render the ground plane using an orthographic projection and either an oblique or perspective projection for the landmarks.

#### Oblique Projection

As shown Figure 5.1, the oblique projection is defined by two parameters $\beta$ and $\alpha$ that specify the direction of the projector lines. In our application the image plane is parallel to the ground plane and $\beta$ controls the amount of foreshortening of the building facades, while $\alpha$ controls which facades of the buildings are visible. Because the image plane is orthogonal to the ground plane the buildings remain correctly orientated with respect to the streets – the oblique projection does not change the orientation of the building footprint.

We build a different oblique projection for each building such that the building facades closest to the nearby roads are visible. When more than two facades face a street, we favor the facades facing the most important streets. More specifically, we compute an orientation vector $\mathbf{d}$ for each building that represents the direction to the closest roads. As shown in Figure 5.1 we subsample the building footprint to form the points $f_i$ for $i = 1...M$ and break the block boundary into a set of segments $\mathbf{s}_j$ for $j = 1..N$ with normals $\mathbf{n}_j$. Then

$$\mathbf{d} = \frac{1}{MN}\sum_{i}^{M}\sum_{j}^{N} W(f_i, \mathbf{s}_j)\mathbf{n}_j$$

is a weighted sum of the segment normals. The weight is given by

$$W(f_i, s_j) = k_j\left(1 - \frac{D(f_i, \mathbf{s}_j)}{\max D(f, \mathbf{s})}\right),$$

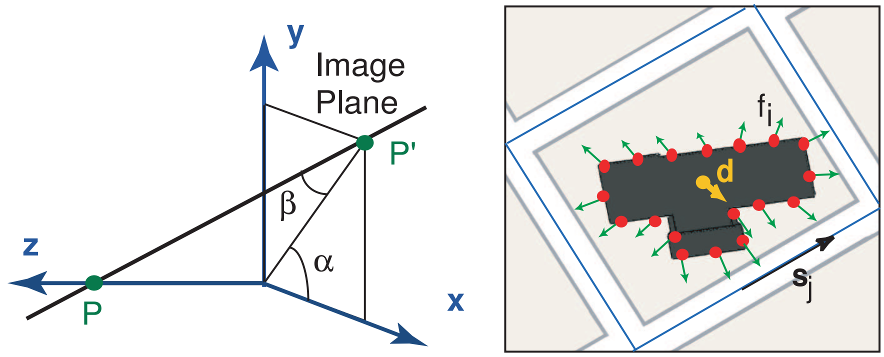



*Figure 5.1: (left) Oblique projection parameters. The projection angle $\beta$ is the angle between the projector line and the image plane. The orientation angle $\alpha$ is the angle in the image plane between the x-axis and the projector. (right) Computing the orientation vector* **d**. *For each footprint point $f_i$ we compute a weighted sum of all the block segment normals where the closest segments are given higher weight (green vectors). Then we sum these vectors to obtain* **d**.

where $k_j$ is the importance score of the road segment $\mathbf{s}_j$ is part of (see Section 4.2 for road importance), $D(f_i, \mathbf{s}_j)$ is the distance between the footprint point $f_i$ and the block boundary segment $\mathbf{s}_j$, and $\max D(f, \mathbf{s})$ is computed over all footprint points and boundary segments. Thus, boundary segments that are closest to the footprint points are given the highest weights.

Finally we use **d** to compute the projection parameters as

$$\alpha = \arctan \frac{d_y}{d_x} \quad \textit{and} \quad \beta = \frac{\pi}{2} - c\|\mathbf{d}\|.$$

We set $\alpha$ to the orientation of **d** so that the street-side building facades are visible. We set $\beta$ proportional to the length of **d** and use the scale factor $c$ to control the amount of foreshortening.

#### Perspective Projection

Rendering the buildings using a perspective projection provides better depth cues and less distortion than using an oblique projection. However, the perspective projection also creates misalignments in translation, rotation and scale between the buildings and the roads that are rendered using the top-down orthographic projection of the ground plane.

To correct these misalignments we first identify the road segment that each building is most aligned with. We initially work in world space and project each vertex of the building footprint geometry onto each road segment adjacent to the building. We then choose the road segment **s** with the longest range of projected points (see Figure 5.2(left)). Next we compute a vector **b** that describes the orientation of the building with respect to **s**. We fit a bounding box oriented in the same direction as **s** to the building footprint. Then, **b** is the edge of the bounding box closest to **s**.

Suppose $P_B$ and $P_G$ designate the building projection matrix and the ground projection matrix respectively and $x$ is the midpoint of **b**. To correct for translational misalignment, we compute a translation matrix $T$ in building space such that $P_B T x = P_G x$, and apply $T$ to all vertices of the building geometry.

To correct for rotational misalignments we compute a rotation matrix $R_z$ about the z-axis of building space such that $P_B R_z \mathbf{b}$ and $P_G \mathbf{b}$ are aligned. Note that we cannot set $R_z = P_B^{-1} P_G$ because in the general case $P_B^{-1} P_G$ will not be a rotation about the z-axis. As shown in Figure 5.2, our approach is to construct a plane in world space that passes through the center of projection ($COP$) of $P_B$ and the point $x$, and intersects the image plane parallel to $P_G \mathbf{b}$. To build the plane we compute a world space vector $\mathbf{b}'$ such that $P_B \mathbf{b}' = P_G \mathbf{b}$, by taking two points on $P_G \mathbf{b}$, giving them the same z coordinate and then transforming them through $P_B^{-1}$ to put them in world space. Then the normal to the plane is given by $\mathbf{n} = (x - COP) \times \mathbf{b}'$. Since this plane passes through the $COP$ any edge that lies in it must be parallel $P_G \mathbf{b}$. Thus, we construct the rotation matrix $R_z$ that rotates $\mathbf{b}$ about the point $x$ such that $\mathbf{b}$ lies in the plane. That is, $R_z \mathbf{b} \cdot \mathbf{n} = 0$. Solving for the rotation angle $\theta$ we obtain

$$\theta = \arcsin \frac{-b_z n_z}{\sqrt{c^2 + d^2}} - \arctan \frac{d}{c},$$

$$\textit{where} \quad c = b_y n_x - b_x n_y \quad \textit{and} \quad d = b_x n_x + b_y n_y.$$

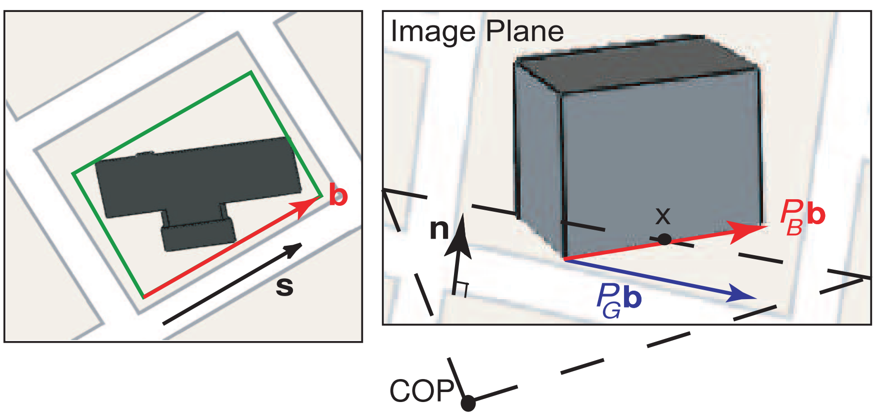



***Figure 5.2:*** *(left) The building footprint is best aligned to street segment* **s***. We compute a bounding box for the footprint in the direction of* **s** *and then identify the bounding box edge* **b** *closest to* **s***. (right) To rotate the buildings we first construct the plane in world space that passes through the center of projection COP of $P_B$ and the point x, and intersects the image plane parallel to $P_G$***b**

After rotating the buildings to re-align them with the roads we scale down buildings if their footprint is larger than their block area. We apply $P_B$ to the building footprint vertices and $P_G$ to the block boundary vertices. Once all vertices are in image space, we perform a binary search to find the scale factor that will reduce the area of the building footprint to fit within the block area. We scale the buildings about $x$, because we know that $x$ is at the appropriate distance from the block boundaries.

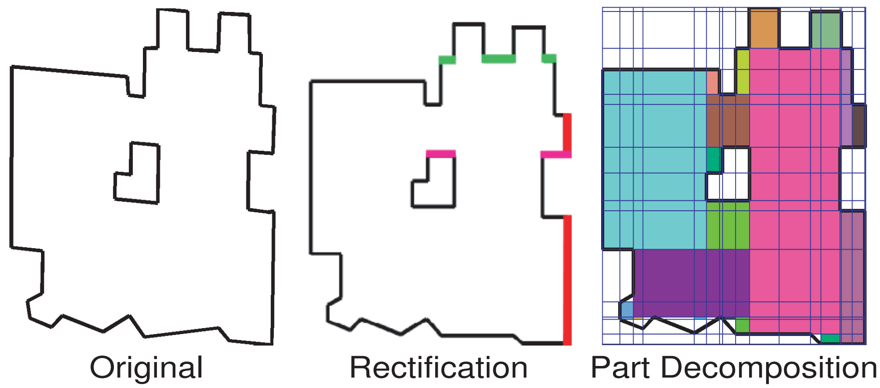



*Figure 5.3:* *Building Simplification. (1) The original footprint. (2) The building footprint after rectification. Colored edges indicate the groups of edges treated identically while rectifying the building. (3) The building footprint after fitting a grid to it. Using a 2D flood flow, we fit rectangles to the full cells.*

## 5.2 Generalization

We use a variety of cartographic generalization techniques including simplification, typification, displacement, deformation and selection to improve the clarity of the map and to emphasize the most important information, while preserving spatial relationships between map objects. We focus on generalization operations applied to landmarks and only use simplification and selection for the roads. We simplify roads by averaging the orientation of two nearly aligned consecutive road segments and select roads by only displaying the ones with highest importance.

#### Building Simplification

Mapmakers often use simplification as a tool to direct a viewer's attention to buildings with more geometric detail by reducing the visual complexity of less important buildings. While automated building simplification is a well-studied area of cartography, most approaches focus on simplifying 2D building footprints [32, 28]. More recent techniques for simplifying buildings in 3D are designed to find and eliminate small volumetric features such as protrusions on the surface model [15, 34, 10]

Our simplification algorithm is based on the method of Forberg [10] who focuses on the the problem of simplifying buildings composed of axis-aligned orthogonal planes. She searches for planar building facets that are parallel and located near one another and then shifts these facets towards each other until they lie on a single plane. We extend this work to handle some non-orthogonal buildings. Our approach proceeds in 3 phases; *rectification, part decomposition* and *facet shifting*. We assume that the buildings are vertical extrusions of their footprint and therefore most of our processing is done on the 2D building footprints.

***Rectification.*** The rectification phase is designed to orthogonalize buildings by reorienting their walls to the principal orientations of the building footprint. To compute the principal axes we bin the building footprint edge segments by their orientation. We

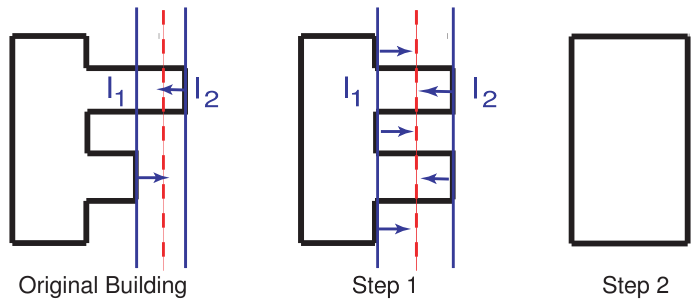



***Figure 5.4:*** *Facet Shifting. In each step the closest parallel facets of the building, $l_1$ and $l_2$, are shifted to lie on the same line*

then identify the bin with the largest number of segments weighted by their length and fit a box to the building with the average orientation of the segments in the bin. All subsequent phases treat the principal axes as the x- and y-axes of the building. We further rectify building footprint edges that are almost parallel (i.e. within 10°) to the principal axes by snapping them to these axes. Similarly we group together parallel footprint edges that are near one another and clamp them to lie on the same line in order to better preserve the structure of the building during simplification.

***Part decomposition.*** We decompose the building into rectangular sub-parts by first embedding it in a non-uniform grid. We generate grid cell boundaries along the principal x-, y- and z-axes at every footprint and roof vertex in the model. We then classify the cells as being fully inside, partially inside, or outside the building. To classify a cell we uniformly subsample it and check whether each sample lies inside the building using a point in polyhedron test. If more than 5% of the samples are inside the building it is marked as partially inside and if more than 95% are inside it is marked as fully inside. Finally, we group together grid cells marked as fully inside into larger rectangular subparts using a greedy flood-fill algorithm.

***Facet shifting.*** To simplify buildings we iteratively apply Forberg's [10] facet shifting algorithm. We first identify the two consecutive grid lines in our part decomposition, $l_1$ and $l_2$, with the smallest spacing in either the x or y direction. As shown in Figure 5.4, we then replace these two lines by a new line at the average x or y location and shift the corresponding facets to this new line. In most cases this process causes the boxes delimited by $l_1$ and $l_2$ to disappear and increases the size of neighboring boxes.

At each iteration of this algorithm we quantify the amount of simplification as the volume of the model eliminated by facet shifting normalized by the volume of the initial, unsimplified orthogonal model. Figure 5.5 shows the result of iteratively simplifying an example building and Figure 5.9 shows simplification applied to a larger area.

This algorithm cannot simplify buildings with significantly inclined roofs or walls such as San Francisco's Transamerica pyramid. However, non-rectangular buildings are uncommon in most cities. In our San Francisco data set, 7% of the 6227 buildings are non-rectangular and remain unsimplified. Moreover, such buildings tend to be visually distinctive and therefore should be treated as important visual landmarks that are rendered without simplification.

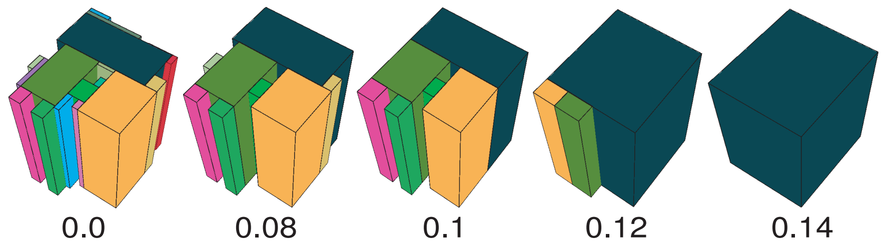



***Figure 5.5:*** *Five steps of building simplification. The coefficients below each step indicate the relative amount of simplification.*

We identify non-rectangular buildings by computing the volume of the partially inside cells over the total building volume. If this measure is below 10%, we drop the partially inside cells and treat the building as orthogonal. Otherwise we mark the building as irregular and do not simplify it. Thus, our approach orthogonalizes buildings with small deviations from orthogonality, but does not simplify large irregular buildings.

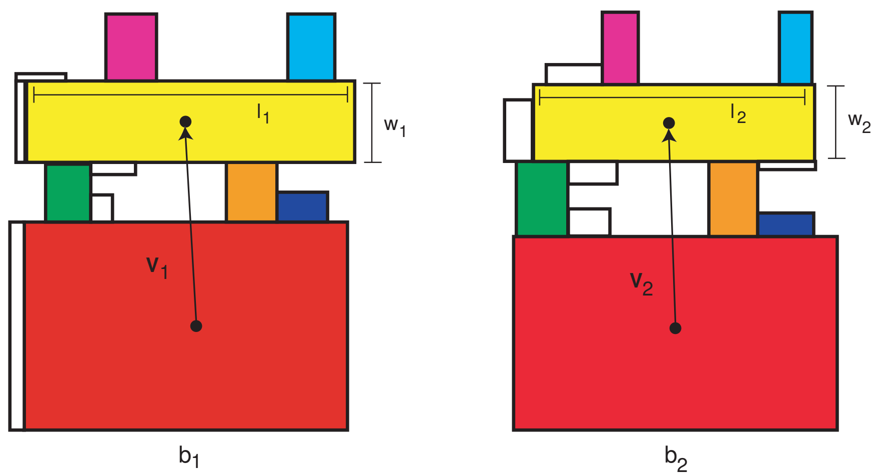



***Figure 5.6:*** *Building Typification. For a building $b_1$ to be similar to a building $b_2$, we iteratively consider each part box of $b_1$ and search for a part box with similar dimensions and position with respect to the new coordinate system in $b_2$. For example, the yellow boxes of $b_1$ and $b_2$ are similar if $|l_1 - l_2| < t$, $|w_1 - w_2| < t$ where $t = 3meters$ and $\frac{v_1}{\|v_1\|} \cdot \frac{v_2}{\|v_2\|} > 0.9$. We draw pairs of matching boxes in the same color. White boxes are too small to be considered and are not matched.*

#### Building Typification

Mapmakers often typify a block by detecting similar structured buildings and drawing them as being identical. This operation increases the clarity of the map, because viewers perceptually group identically looking buildings and the map becomes easier to organize and understand.

We detect similar buildings in a block by comparing their box decomposition. We first determine all the buildings that have approximately the same dimensions by comparing

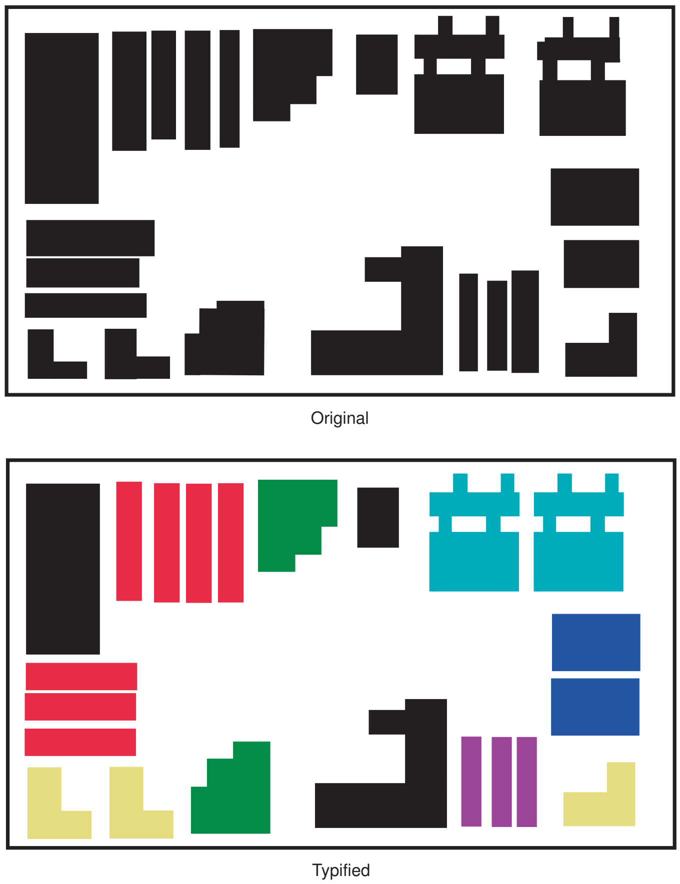



*Figure 5.7:* *Typification applied to a block. The original block (top) and the typified block (bottom) where the sets of identical buildings that result after typification are drawn in the same color. Buildings rendered in black (bottom) were not typified.*

the length, height and width of their bounding box. Then, we sort the boxes in the building part decomposition by their volume. We recenter the part boxes with respect to the coordinate system defined by the center of the largest box and the principal axis of the building. For a building $b_1$ to be similar to a building $b_2$, we iteratively consider each part box of $b_1$ and search for a part box with similar dimensions and position with respect to the new coordinate system in $b_2$. More specifically, we first compare the largest box of $b_1$ to the largest box of $b_2$. If their difference in length, height and width is neglectable, we iteratively consider the next largest box of $b_1$ and search in the box structure of $b_2$ for a matching box until the volume of the box of $b_1$ is below a threshold. Two boxes match if they have approximately the same dimensions and the same relative position. The positions are compared by looking at the orientation of the box center with respect to the origin. The distance to the origin is implicitly compared by checking the dimensions of the boxes at each iteration. Figure 5.6 illustrates the matching criterion and shows all pairs of matching boxes for an example building.

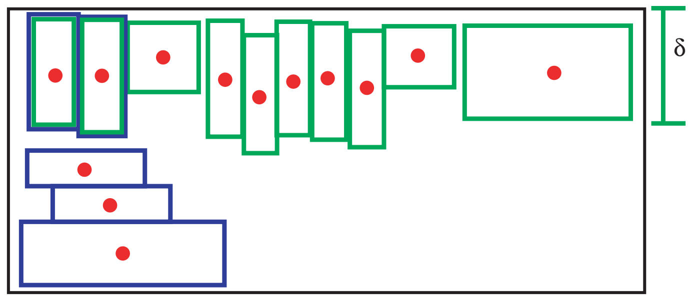



*Figure 5.8:* *Alignment groups. Buildings that are within a distance δ of an edge of a block boundary form an alignment group. The green alignment group is within δ of the top block edge and the blue group is within δ of the left block edge. Buildings like the ones in the top left can belong to more than one alignment group.*

Figure 5.7 visualizes the result of typifying a block.

### Building Layout

Mapmakers often widen roads in order to emphasize the paths in an unknown place and prevent buildings from occluding the roads in dense areas. As a consequence, the space available for buildings in the block is limited. A common approach is to use map generalization to adjust the buildings to fit within the limited space [22, 40, 41, 4, 38].

We follow the approach of Ware and Jones [40] and generalize the buildings within a block using simulated annealing. While their work only treats the displacement of buildings, we include scaling and removal of buildings to allow for greater flexibility in generalization. Like any simulated annealing-based technique we must define three procedures in order to run the optimization: *initialization, perturb* and *scoring*.

***Initialization.*** Widening the roads reduces the size and changes the boundary of each block. Thus, to initialize the optimization, we first determine the new block boundaries by re-triangulating the widened road network and applying a flood-fill over the triangles stopped by the streets. For each building, we determine a new approximate position within the reduced block by computing the orientation vector to the center of the original block and scaling it by the best uniform scale mapping between the original block and the reduced size block. Finally, we compute alignment groups for a block that contain all the buildings aligned to the same street as shown in Figure 5.8.

***Perturb.*** The perturb function for the search randomly picks a building to alter and randomly chooses a generalization operator for that building. We have implemented three operations: displacement of the building parallel or perpendicular to the direction of the alignment group, scaling of the building, removal and reinsertion of the building.

***Scoring.*** The layout scoring function evaluates each building on a number of criteria. More specifically, we penalize a building if (1) it falls outside the block boundary, (2) it moves far away from any street it is aligned with, (3) it falls out of alignment with other buildings in the same alignment groups,(4) its position ordering within its alignment groups changes, (5) it is reduced in scale significantly and it has high importance, (6) it is scaled differently from its neighbors, (7) it overlaps other buildings, (8) it is removed from the map.

If conflicts in generalization, such as building overlaps, cannot be resolved by the annealing process, the resulting map will contain artifacts. In general the system chooses a suitable compromise, yet our automated generalization methods cannot compete with the best works of highly proficient map artists. Nevertheless, we believe that our visualizations surpass in quality many existing tourist maps in use today.

Figure 5.9 shows block generalization applied to an area around Union Square in San Francisco. The simplification coefficients for the buildings are inversely proportional to their importance score (see Section 4.1).

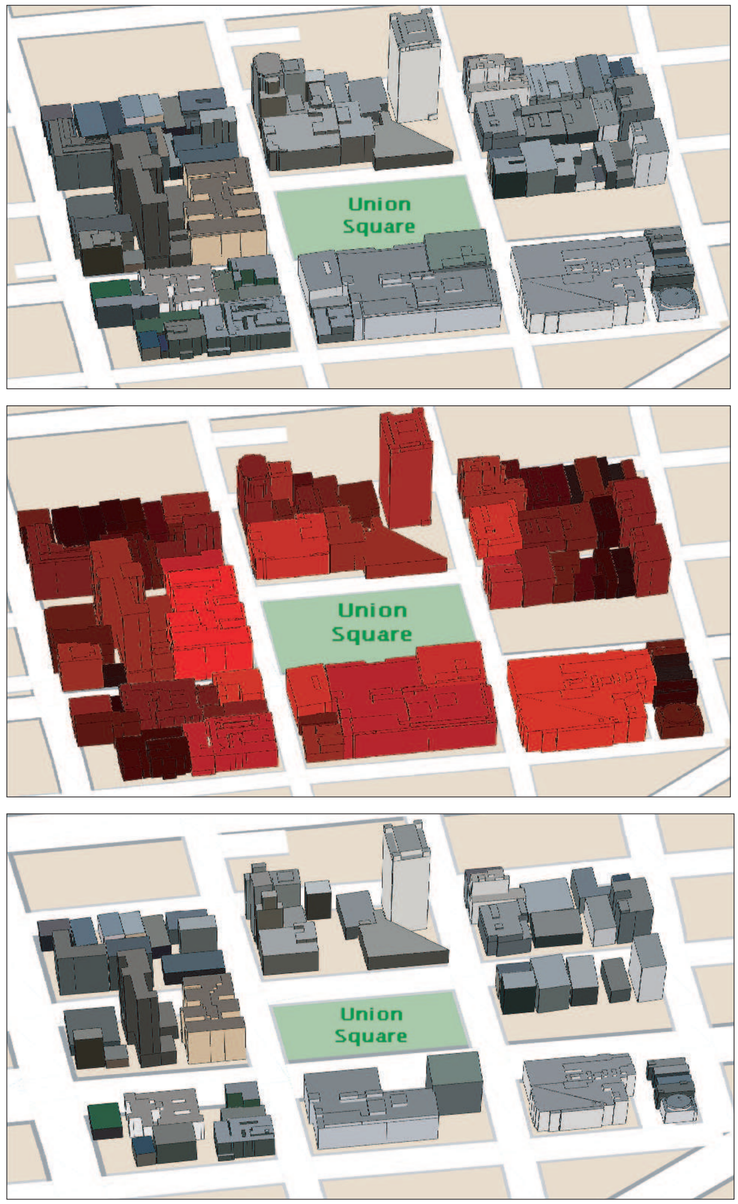



***Figure 5.9:*** *Generalization. (Top) Original block. (Middle) Importance scores for the buildings, where the red saturation of a building is proportional to its importance. (Bottom) Generalized blocks.*

## 5.3 Non-Photorealistic Rendering

Mapmakers often emphasize landmarks by marking their contour lines and attenuating the building color. Contour lines convey information about the shape and help a viewer to better identify the building's geometry. We determine the set of contour lines by identifying the edges between any two neighboring triangles that have a dihedral angle of more than $\pi/4$.

Artifacts in the building texture such as shadows can be very distracting for a viewer and make the building harder to recognize. To counteract this effect, our system can render the walls and the roofs using a uniform color. After extracting the wall and roof colors, as explained in Section 4.1 we further saturate the wall colors to make them more distinctive and we brighten the roof colors to make the buildings appear as if they are lit from above. To de-emphasize non-landmark buildings we render them in light gray.

## 5.4 Labeling

For tourist maps to be usable and to allow navigation, we must label important map elements including landmarks, roads, nodes (lakes, parks, squares) and districts. Automated placement of labels and line features on maps is a well-studied problem in automated map design [13, 8, 1, 39]. A common approach is to search the space of possible labelings of the map to find an optimal layout. We build on the simulated annealing approach of Agrawala and Stolte [1].

We label the map elements in four stages starting with nodes, and then the landmarks, roads and districts in consecutive order. Breaking the search into stages in this manner, reduces the dimensionality of our search space and significantly improves convergence. Our system extends the previous labeling techniques by including the importance values of the map elements in the label layout scoring function. This approach ensures that the most important elements will be labeled in the optimal way, while the labels of less important elements might be placed further away or even removed.

# Chapter 6

# Results and Conclusions

## 6.1 Generated Maps

Examples of several tourist maps generated using our system are presented in Figures 1.1, 6.1 and 6.5. The input data consists of a set of 6227 buildings of downtown San Francisco. We identify 1257 of these as semantic landmarks by parsing three input websites (see Section 4.1).

The maps of Figure 6.1, 6.2, 6.3 and 6.4 show a set of distinctive landmarks in San Francisco for different categories of interest. The first image is a combination of the most important landmarks in San Francisco disregarding their category and the most important roads. Tourists with differing personal interests can adapt the map to show only tourist attraction (Figure 6.2), restaurants (Figure 6.3) and shopping (Figure 6.4). In all of these maps we emphasize points of interest by drawing them in color with dark labels, while other salient buildings and their labels are drawn in light gray. In order to increase the visibility of the landmarks and the ground plane, we render the buildings using a perspective projection and the roads with an orthographic projection. This map only displays the most important roads. The width of a road and the color of its label depend on the road category.

Figure 6.5 shows a close-up view of the buildings along Market Street. We use an oblique projection for the buildings which emphasizes the street-side facades of the buildings. In this case the facades facing Market Street are always visible since this is the street with the highest importance in that area. We label semantic landmarks and de-emphasize other buildings by making them semi-transparent.

Figure 6.6 shows the advantage of widening the roads and generalizing the buildings for the area around Grace Cathedral in San Francisco. In the left image California and

Clay St are not visible. After widening the roads and generalizing the buildings, the two occluded streets become visible while the overall structure of the block is preserved.

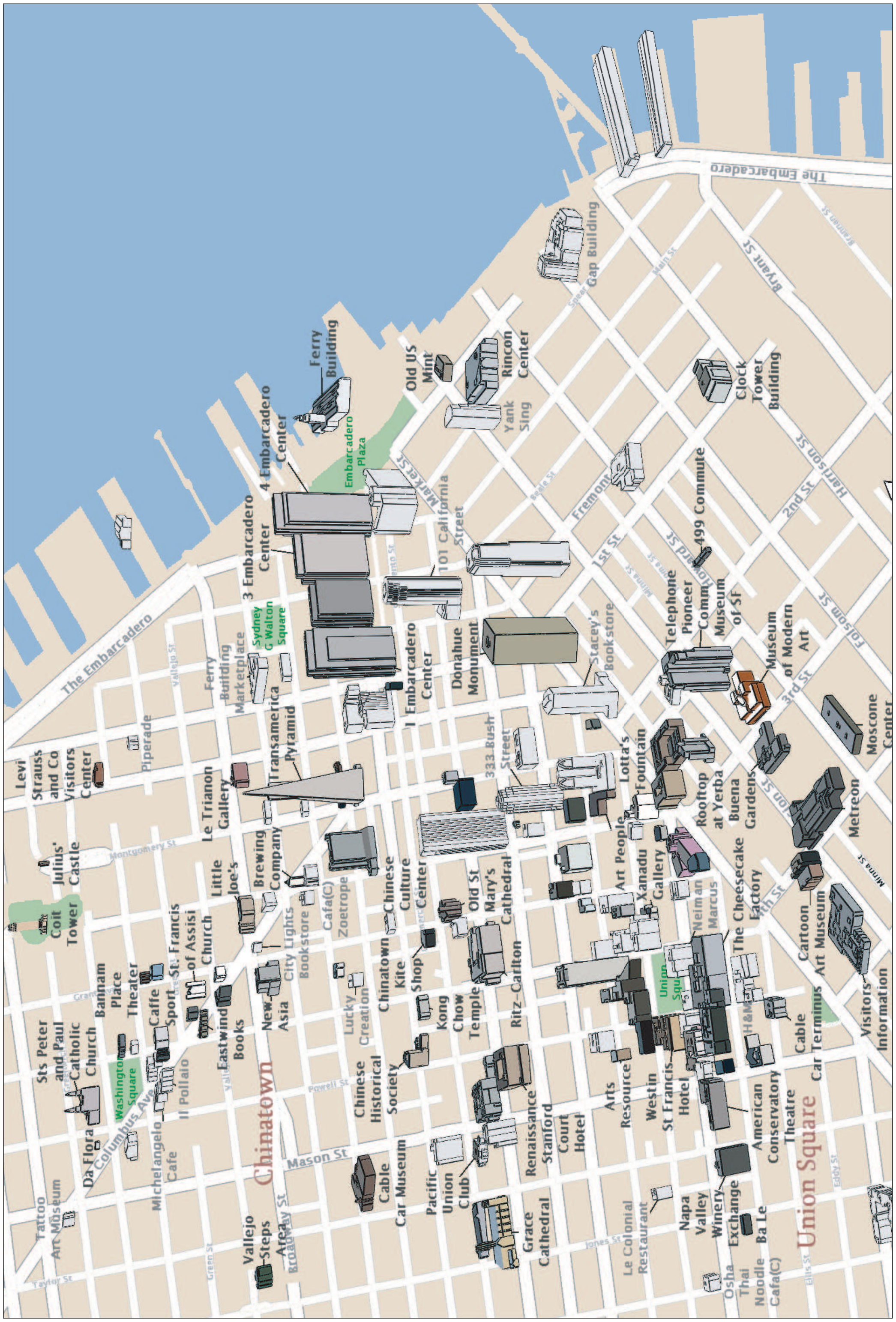



***Figure 6.1:*** *Semantic landmarks in San Francisco. A general-purpose map showing the most important landmarks across all semantic categories.*

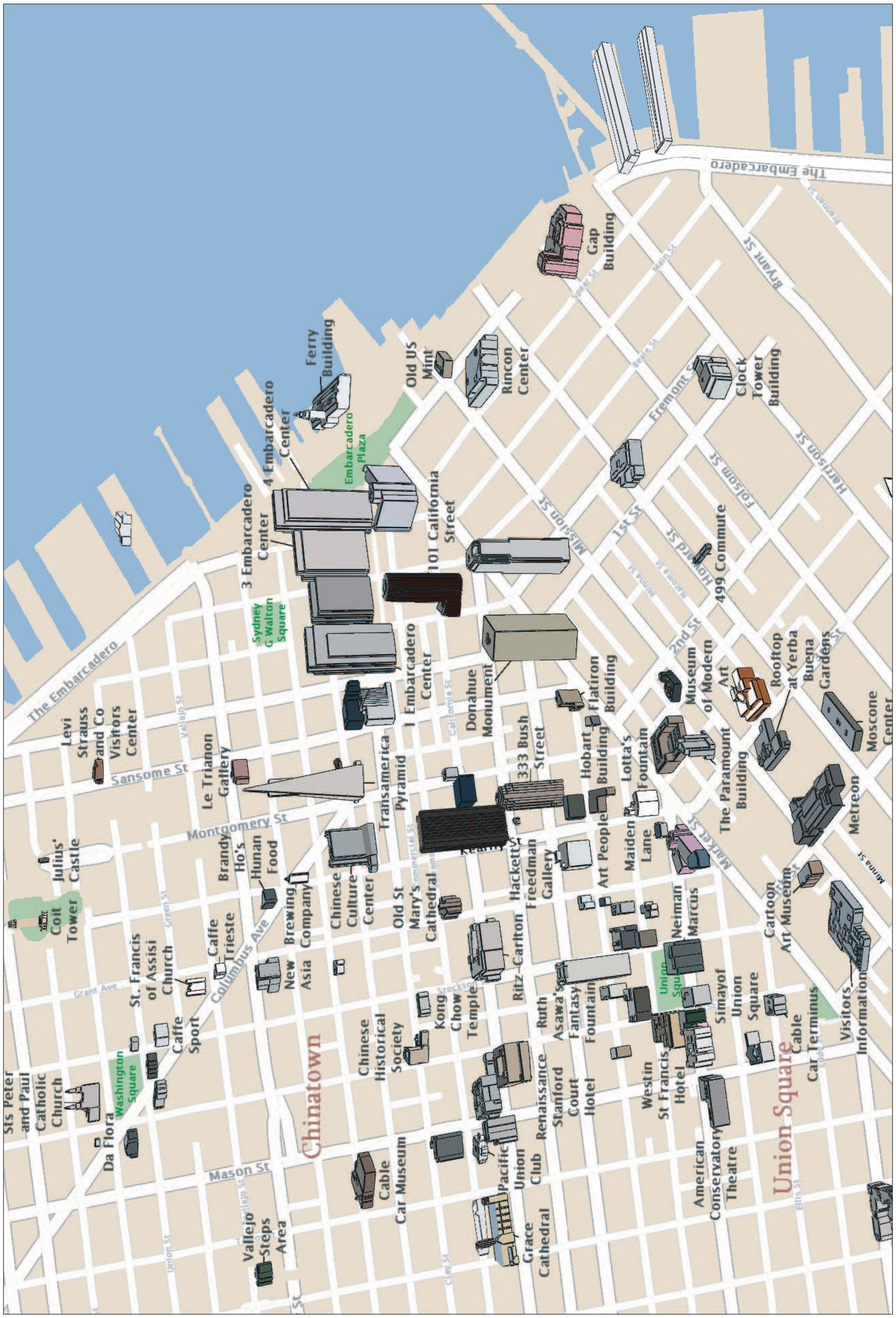



*Figure 6.2: A map that highlights tourist attractions.*

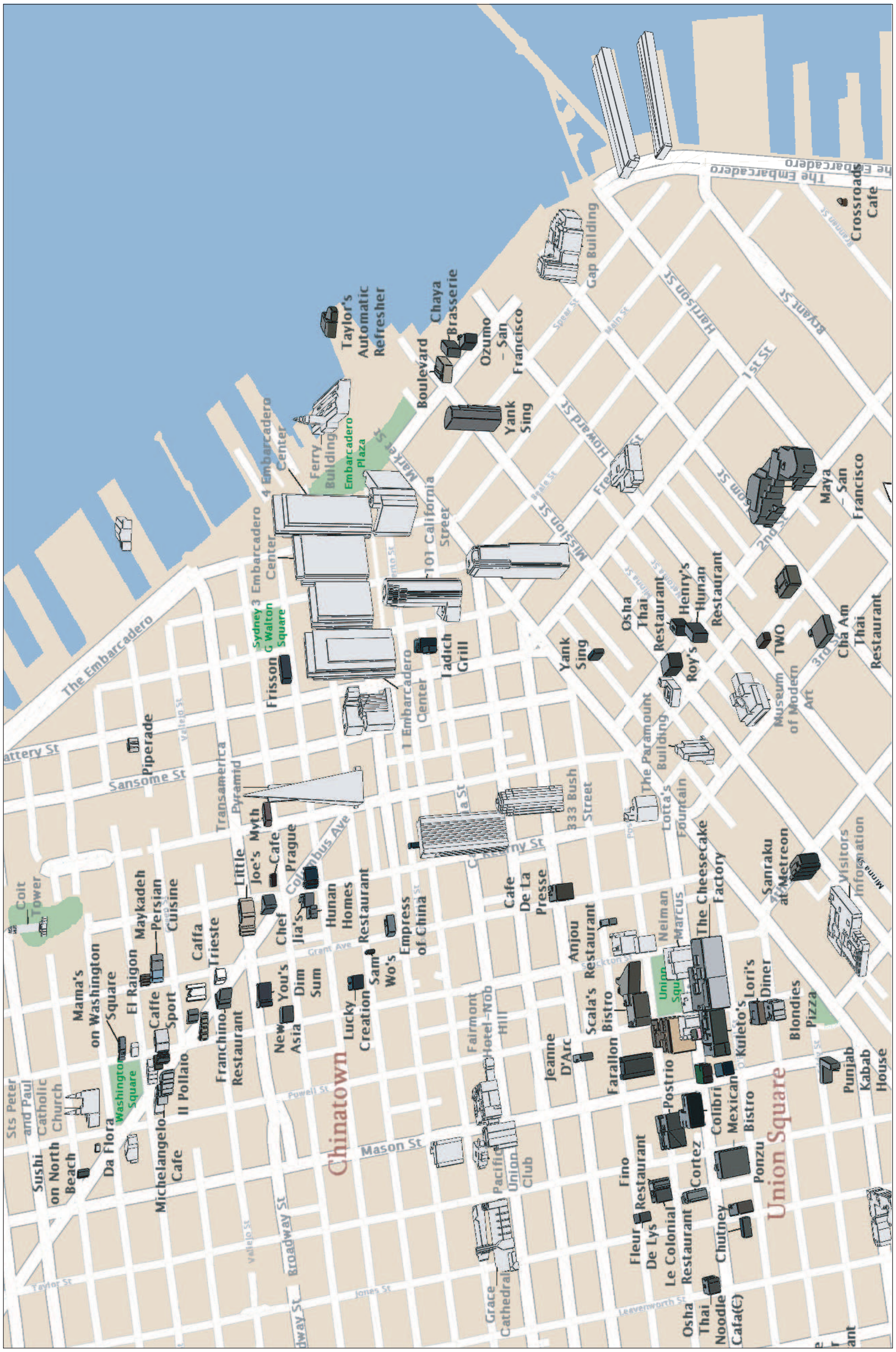



*Figure 6.3:* *A map that highlights restaurants.*

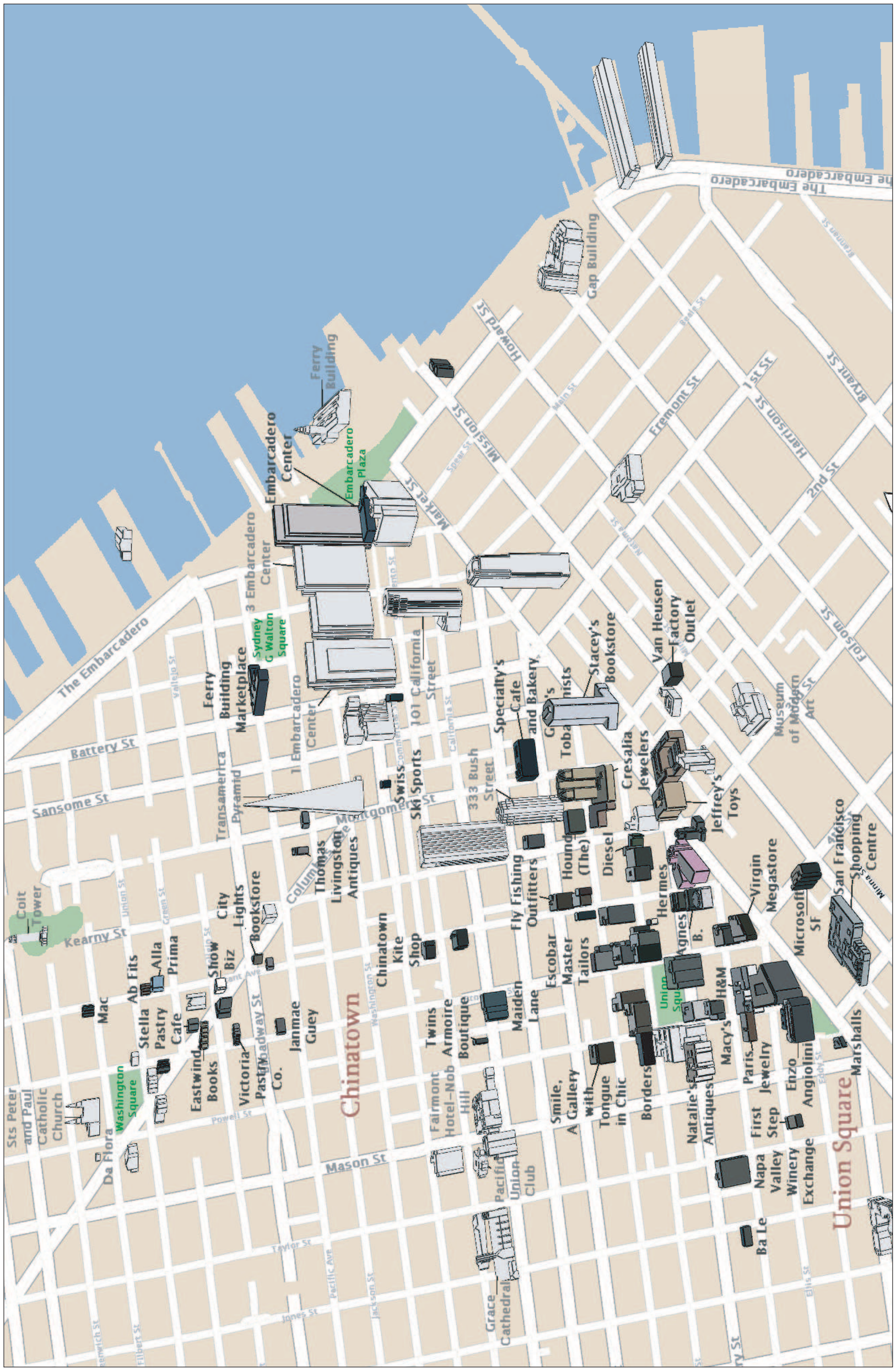



*Figure 6.4:* A map that highlights shopping areas.

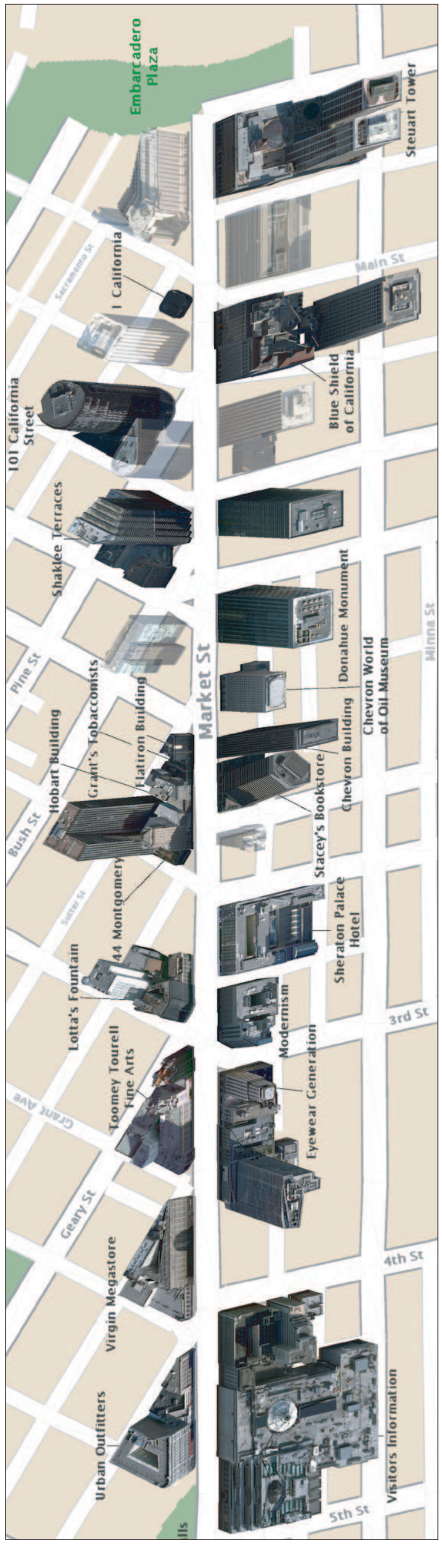



***Figure 6.5:*** *Market Street. Buildings are rendered with an adaptive oblique projection that ensures that street-side facades are visible so that tourists can easily recognize them as they walk down the street.*


***Figure 6.6:*** *Generalization applied on the area around the Grace Cathedral in San Francisco. After widening the streets and generalizing the buildings the occluded California and Clay St in the right image become visible (bottom).*

## 6.2 Comparison with Existing Maps

We compare the tourist maps generated with our approach to existing automated and manual approaches.

Figure 6.7 shows a hand-drawn tourist map of San Francisco. Note that this map is from a different viewpoint than our result maps ( Figure 6.1, 6.2, 6.3 and 6.4). Our maps do not yet match the rendering quality of the best hand-designed maps created by experts and do not include as much information as they do. For example, the hand-designed map of Figure 6.7 includes information about the terrain, this data is not yet available to our system. On the other hand, the hand-made map fails to adapt to the interest of a user. It shows a variety of landmarks that meet the interests of most viewers. As a consequence, it appears very cluttered with buildings occluding each other and labels that are difficult to read. Our maps do not need to display as many landmarks and can emphasize the most relevant ones since they are generated for a particular user. Finally, the maps made by artists cannot adapt to changes in the city, a new map would need to be drawn each time.


***Figure 6.7:*** *An example hand-drawn map for comparison with our results. The hand-designed map is superior to our results in the rendering quality, but fails to adapt to the specific needs of its users. As a consequence it displays a large number of various landmarks and appears cluttered with labels that become difficult to read and buildings occluding each other.*

We also compare our results to existing digital maps such as the ones shown in Figure 1.1 and 6.8. The main advantage of digital maps and our maps is that they are able to adapt to the interests of individual tourists and to changes in the city. Compared to our maps, digital maps include some information such as Bart stops that is not yet available to our system. A disadvantage of digital maps is that they do not render landmarks in a representative way. Figure 6.8 shows only building footprints and marks landmarks by using pushpins. Figure 1.1 shows satellite images of the real buildings that suffer from shadows and other artifacts which makes their shape and color difficult to recognize. We render more recognizable and abstracted 3D views of the landmarks. In addition we use cartographic generalization techniques to reduce the complexity of the map and facilitate navigation. Digital maps render either all the buildings with their complex 3D geometry (Figure 1.1) or all the building footprints without simplification (Figure 6.8).

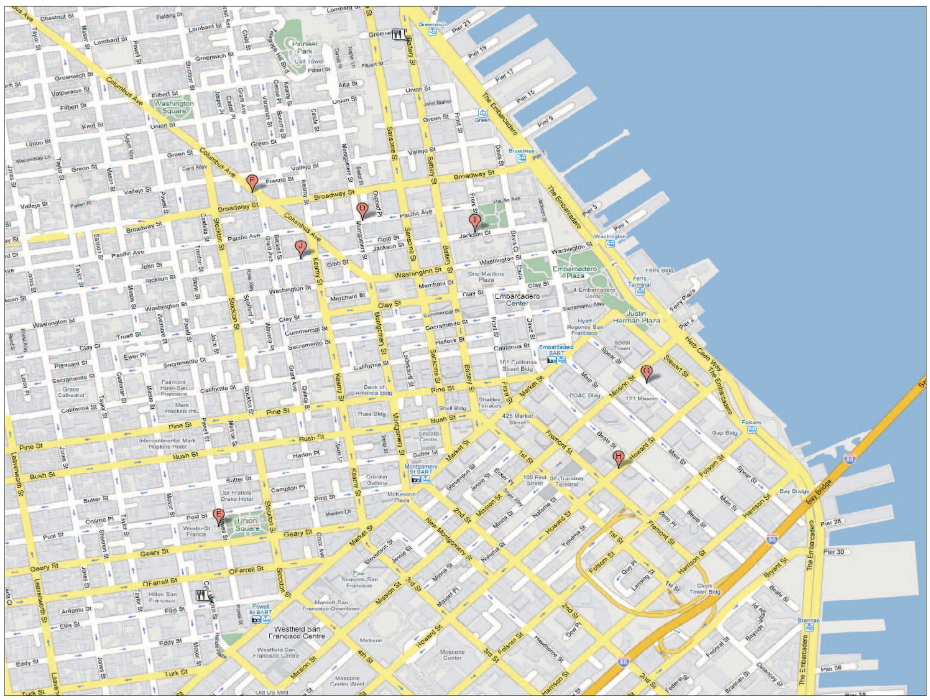


*__Figure 6.8:__ An example of a digital map for comparison with our results. Digital maps include some information such as Bart stops that is not yet accessible to our system. The main disadvantage of this digital map is that it only shows building footprint while we render more recognizable 3D views of landmarks. In addition we use cartographic generalization techniques to reduce the complexity of the map. This digital map does not modify the building footprints.*

## 6.3 Timing Information

Our tourist map generation system is comprised of several precomputation steps that compute most of the semantic, visual and structural features for the buildings and roads. Some features can only be computed in the authoring application since they depend on the landmark categories of interest selected by a user. We also precompute the part rectification and part decomposition of the buildings as well as the road independant weights $\frac{W(f_i,s_j)}{k_j}$ for the oblique projection . Because these computations are performed once, we have not focused on optimizing them and most of them are implemented in Matlab.

While most of the precomputation steps take on the order of minutes to process the 6227 buildings in our data set a few take much longer. The slowest precomputation step is computing the route between each pair of landmarks (structural road feature) because we rely on the Virtual Earth web-based API and querying the routes is slow. This step takes about 24 hours to compute for all pairs of semantic landmarks. We provide more specific timing in table 6.3.

| | **Variable** |
|---|---|
| Number of Buildings | $b$ |
| Number of rendered Buildings (user dependant) | $b_r$ |
| Number of Semantic Landmarks | $l$ |
| Number of Points of Interest (user dependant) | $p$ |
| Number of Building Triangles | $t$ |
| Number of Road Segments | $r$ |
| Number of Road Intersections | $i$ |
| Number of Squares | $s$ |
| Number of Nodes | $n$ |
| Size of the Ground Texture | $g$ |
| Maximal Number of Footprint Edges for a Building | $f$ |

***Table 6.1:*** *Definition of the variables used in the timing tables.*

In a second step our map authoring application loads this precomputed data and allows users to modify importance weights, set the viewing parameters and the landmark categories of interest. To display a mpa we generalize the buildings, perform multi-perspective rendering and then label the map. Generalizing a single blockof buildings can take between a few seconds (if the block contains very few landmark buidings) up to a few minutes (if the block contains many landmark buildings). Multi-perspective rendering occurs in real-time. Labeling is slow largely because we have not implemented and specialized data structures to speed up the spatial overlap computations. A map containing hundreds of labels such as the on in Figure 6.1 can take tens of minutes to label. With a small number of labels as in Figure 6.5, labeling takes tens of seconds. However, we believe that using the appropriate data structures as well as GPU accelerations we can significantly reduce labeling time by 1 or 2 orders of magnitude. We summarize specific timings for the authoring application in table 6.3.

| Stages | | | | Complexity | Timing |
|---|---|---|---|---|---|
| ***Computing Importance*** | | | | | |
| | Landmarks | | | | |
| | | Semantic Features | | $O(l)$ | $22min$ |
| | | Visual Features | | | |
| | | | Facade Color | | * |
| | | | Shape Complexity | $O(t)$ | $9min$ |
| | | | Building Height | $O(b)$ | $1sec$ |
| | | Structural Features | | | |
| | | | Buildings around squares | $O(b \cdot s)$ | $3sec$ |
| | Roads | | | | |
| | | Semantic Features | | | |
| | | | Categories | – | $0sec$ |
| | | | Traffic | $O(r)$ | $7min$ |
| | | Structural Features | | | |
| | | | Roads connecting Landmarks | $O(l^2)$ | * |
| | Districts | | | | |
| | | Semantic Features | | $O(l)$ | $5sec$ |
| | Nodes | | | | |
| | | Semantic Features | | | |
| | | | Squares | $O(r \ln r)$ | $3min$ |
| | | | Park and Lakes | $O(n \cdot g)$ | $17min$ |
| | | | (includes downloading the texture tiles from the internet) | | |
| ***Generalization*** | | | | | |
| | Simplification | | | | |
| | | Rectification | | $O(b \cdot f^2)$ | $5h$ |
| | | Part Decomposition | | | * |
| ***Multiperspective Maps*** | | | | | |
| | Oblique Projection | | | | |
| | | Weight Computation | | $O(f \cdot r)$ | $40min$ |

* *between* $10h$ *and* $1day$

***Table 6.2:*** *Timing. We list the complexity of each of the preprocessing stages of our system and indicate how long each stage takes in minutes.*

| Stages | Complexity | Timing | | |
|---|---|---|---|---|
| | | Figure 6.1 | Figure 6.5 | Figure 6.6 |
| ***Computing Importance*** | | | | |
| (computed for all the buildings once after the landmark categories of interest have been set) | | | | |
| Roads | | | | |
| Structural Features | | | | |
| Landmark Proximity | $O(p \cdot r)$ | 1*sec* | 1*sec* | 1*sec* |
| Landmarks | | | | |
| Structural Features | | | | |
| Buildings at Intersections | $O(b \cdot i)$ | 9*sec* | 9*sec* | 9*sec* |
| ***Generalization*** | | | | |
| (only computed for the rendered buildings) | | | | |
| Simplification | | | | |
| Facet Shifting | | 8*sec* | – | 17*sec* |
| Building Layout | *indetermined** | 25*sec* | 10*sec* | 40*min* |
| ***Multiperspective Maps*** | | | | |
| (only computed for the rendered buildings) | | | | |
| Oblique Projection | $O(r \cdot f \cdot b_r)$ | – | 6*sec* | $<$ 1*sec* (non-adaptive oblique projection) |
| Perspective Projection | $O(b_r)$ | 2*sec* | – | – |
| ***Labeling*** (only computed for the rendered buildings and streets) | *indetermined** *NP – Complete Problem* | 40*min* | 11*min* | 5*sec* |

* (*not guaranteed to converge*)

***Table 6.3:*** *Timing. We list the complexity of each of the stages of the authoring application and indicate how long each stage takes in minutes.*

## 6.4 Conclusions

We have introduced a comprehensive system for automatically generating tourist maps. One of the greatest appeals of approach is customization. Users can easily build their own maps according to their preferences and intentions for a particular visit to a city. In contrast to hand-drawn maps, our system makes use of up-to-date information and could thus even accommodate dynamic information such as road closures or specific opening hours of museums or restaurants. Our system exemplifies how a combination of web-based data mining, perception-guided geometry and image analysis, and advanced visualization techniques yields an effective tool for the automatic creation of tourist maps suitable for everyday use.

# Bibliography


[1] M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249, 2001.

[2] M. Agrawala, D. Zorin, and T. Munzner. Artistic multiprojection rendering. *Eurographics Rendering Workshop 2000*, 2, 2000.

[3] D. Appleyard. Why buildings are known: A predictive tool for architects and planners. *Environment and Behavior*, 1(2):131, 1969.

[4] M. Bader. *Energy Minimization Methods for Feature Displacement in Map Generalization*. PhD thesis, University of Zurich, 2001.

[5] A.K. Deakin. Landmarks as navigational aids on street maps. *Cartography and Geographic Information Science*, 23(1):21–36, 1996.

[6] M. Denis, F. Pazzaglia, C. Cornoldi, and L. Bertolo. Spatial discourse and navigation: an analysis of route directions in the city of venice. *Applied Cognitive Psychology*, 13(2):145–174, 1999.

[7] D. DiBiase, A.M. MacEachren, J.B. Krygier, and C. Reeves. Animation and the role of map design in scientific visualization. *Cartography and Geographic Information Systems*, 19(4):201–214, 1992.

[8] S. Edmondson, J. Christensen, J. Marks, and S.M. Shieber. A General Cartographic Labelling Algorithm. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 33(4):13–24, 1996.

[9] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[10] A. Forberg. Generalization of 3d building data based on a scale-space approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(2):104–111, 2007.

[11] J. Hays and A.A. Efros. Scene completion using millions of photographs. *International Conference on Computer Graphics and Interactive Techniques*, 2007.

[12] N. Holmes. *Pictorial Maps*. Watson-Guptill, 1991.

[13] E. Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[14] B. Jiang and C. Claramunt. A structural approach to the model generalization of an urban street network. *GeoInformatica*, 8(2):157–171, 2004.

[15] M. Kada. Automatic generalisation of 3d building models. *Joint International Symposium on Geospatial Theory, Processing and Applications? 02*, 2002.

[16] A. Klippel and S. Winter. Structural salience of landmarks for route directions. *Spatial Information Theory: International Conference, COSIT 2005*, 2005.

[17] M.J. Kraak and F.J. Ormeling. Cartography. visualization of spatial data. *Harlow: Longman*, 1996.

[18] K.L. Lovelace, M. Hegarty, and D.R. Montello. Elements of good route directions in familiar and unfamiliar environments. *In Proc. COSIT '99*, 1999.

[19] K. Lynch. *The Image of the City*. MIT Press, 1960.

[20] A.M. MacEachren. *How Maps Work: Representation, Visualization, and Design*. Guilford Pubn, 1995.

[21] W.A. Mackaness and K.M. Beard. Use of graph theory to support map generalization. *Cartography and Geographic Information Science*, 20(4):210–221, 1993.

[22] R.B. McMaster and K.S. Shea. *Generalization in digital cartography*. Association of American Geographers, 1992.

[23] P.E. Michon and M. Denis. When and why are visual landmarks used in giving directions? *In Proc. COSIT '01*, 2001.

[24] B. Mirtich. Fast and Accurate Computation of Polyhedral Mass Properties. *Graphics Tools: The JGT Editors' Choice*, 2005.

[25] M. Monmonier. *Mapping It Out: Expository Cartography for the Humanities and Social Sciences*. University Of Chicago Press, 1993.

[26] C. Nothegger, S. Winter, and M. Raubal. Computation of the salience of features. *Spatial Cognition and Computation*, 4(2):113–136, 2004.

[27] M. Raubal and S. Winter. Enriching wayfinding instructions with local landmarks. *GIScience: Second International Conference*, 2002.

[28] M. Sester. Generalization based on least squares adjustment. *International Archives of Photogrammetry and Remote Sensing*, 33:931–938, 2000.

[29] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006.

[30] M.E. Sorrows and S.C. Hirtle. The nature of landmarks for real and electronic spaces. *In Proc. COSIT '99*, 1999.

[31] M. Southworth and S. Southworth. *Maps: A Visual Survey and Design Guide*. Little Brown and Company, 1982.

[32] W. Staufenbiel. *Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung grossmasstäbiger Gebäudedarstellungen*. PhD thesis, Hanover., 1973.

[33] T. Tezuka and K. Tanaka. Landmark extraction: a web mining approach. spatial information theory. *Lecture Notes in Computer Science*, 3693:379–396, 2005.

[34] F. Thiemann. Generalization of 3d building data. *Geospatial Theory, Processing*

*and Applications", IAPRS*, 34(Part 4), 2002.

[35] RC Thomson and DE Richardson. A graph theory approach to road network generalisation. *Proceedings of the 17th International Cartographic Conference*, pages 3–9, 1995.

[36] RC Thomson and DE Richardson. The ?good continuation?principle of perceptual organization applied to the generalization of road networks. *Proceedings of the 19th International Cartographic Conference, Ottawa*, pages 1215–1223, 1999.

[37] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[38] M. van Kreveld. Smooth generalization for continuous zooming. *Proc. 20th Intl. Geographic Conference*, pages 2180–2185, 2001.

[39] I. Vollick, D. Vogel, M. Agrawala, and A. Hertzmann. Specifying label layout style by example. *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 221–230, 2007.

[40] J.M. Ware and C.B. Jones. Conflict reduction in map generalization using iterative improvement. *GeoInformatica*, 2(4):383–407, 1998.

[41] R. Weibel and C.B. Jones. Computational perspectives on map generalization. *GeoInformatica*, 2(4):307–314, 1998.

[42] A. Zipf. User-adaptive maps for location-based services (lbs) for tourism. *Information and Communication Technologies in Tourism*, 2002.