

Automatic Generation of Destination Maps

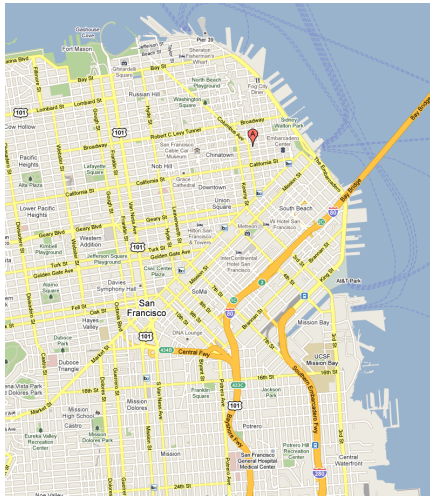
Johannes Kopf
Microsoft Research

Maneesh Agrawala
UC, Berkeley

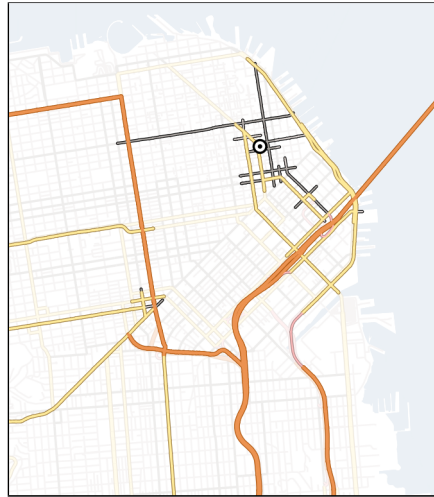
David Barger
Microsoft

David Salesin
Adobe Systems, Inc.

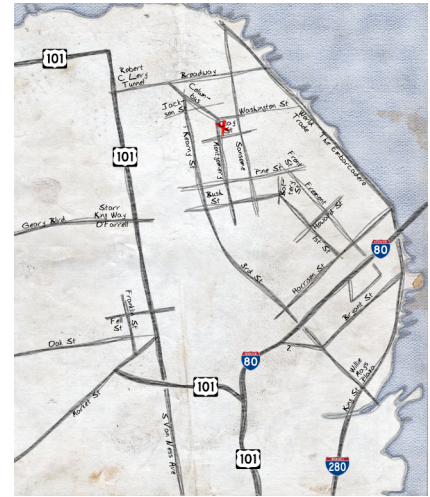
Michael Cohen
Microsoft Research



General Purpose Online Map



Selected Roads



Our Result

Figure 1: A general purpose online map (from Google Maps) with fixed scale is not an effective destination map because the neighborhood streets disappear. Our system selects a relevant subset of the highways, arterials and residential roads required to reach the destination. It then lays out the selected roads, so that all the selected roads are visible and renders the map in a hand-drawn style.

Abstract

Destination maps are navigational aids designed to show anyone within a region how to reach a location (the destination). Hand-designed destination maps include only the most important roads in the region and are non-uniformly scaled to ensure that all of the important roads from the highways to the residential streets are visible. We present the first automated system for creating such destination maps based on the design principles used by mapmakers. Our system includes novel algorithms for selecting the important roads based on mental representations of road networks, and for laying out the roads based on a non-linear optimization procedure. The final layouts are labeled and rendered in a variety of styles ranging from informal to more formal map styles. The system has been used to generate over 57,000 destination maps by thousands of users. We report feedback from both a formal and informal user study, as well as provide quantitative measures of success.

1 Introduction

You are planning a party at your house for 50 friends and acquaintances. Although they all live within a 25 mile radius of your house,

many of them haven't visited your home before. You plan to send them a map showing them how to get to your house. What is the best way to create a map that will accommodate all your guests?

This problem arises whenever many people from a region need to navigate to a specific destination. Invitations to group events such as parties, weddings, and yards sales often include such *destination maps* which show only the subset of roads along all the major routes to the location. Businesses such as restaurants, shops, museums, theaters and sales offices commonly print such destination maps in advertisements, yellow-pages and on the backs of business cards.

To date, such destination maps have always been created manually. Well designed destination maps use cartographic generalization techniques including distortion, simplification, and abstraction to emphasize the most important roads in a map. Yet, creating an effective destination map by hand is a time-consuming process that requires expert map design skills. As a shortcut, many people simply mark the destination on a general-purpose online map (Figure 1 left). Although such maps are precise and show all roads in the area, they are difficult to use because they do not generalize the map to emphasize the roads necessary to reach the destination. This is especially true in the area around the destination, which is often too small to understand how to navigate the last few blocks to the destination.

Mapmakers face two challenges when designing a destination map:

Road selection. The first challenge is to select the relevant subset of the highways, arterials, and residential roads that are directly on-route to the destination, while eliminating extraneous roads that are unlikely to help navigators reach the destination. Destination maps are usually designed for people who are familiar with the overall region, but unfamiliar with the area right around the destination. Therefore, these maps usually include the local roads near the destination and progressively reduce detail, including only the larger

roads out to the surrounding highways. In contrast, because online maps are general-purpose maps and are not designed for navigating to a specific destination, they must include all of the roads in the region, and they are often cluttered with irrelevant detail.

Road layout. The second challenge is to lay out the selected network so that all of the roads are visible. In order to follow a map to the destination, navigators must be able to see all of the turning points and associated roads in the map. To ensure such visibility, mapmakers often simplify and then rescale the roads non-uniformly to increase the length of short neighborhood roads, while decreasing the length of long highways. In contrast, most online maps use a fixed scale for the entire map. As a result it is either impossible to see the short roads because they shrink to a point, or to see the long roads because they fall outside the map. One notable exception to this approach is LineDrive [Agrawala and Stolte 2001], which renders point-to-point driving directions in a schematized style and non-uniformly scales roads to ensure good visibility for all of them.

In this paper, we present the first automated system for generating destination maps to address both of these challenges. The user specifies a destination and region of interest as input, and our system selects the relevant network of roads using heuristics based on the way people mentally think about and use maps. Our system then lays out the network in a sequence of three steps; first simplifying the geometry of the selected roads; then, optimizing the position, scale, and orientation of roads non-uniformly to provide good visibility for all of them; and finally adding in geographic contextual information such as bodies of water located in the area. The resulting maps are similar to hand-designed destination maps. They significantly reduce the problems of extraneous clutter and fixed scale intrinsic to the general-purpose online maps. Figure 1 shows an example of an online map, the subset of roads selected by our system, and the final destination map produced by our system.

2 Design Principles

Well-designed destination maps emphasize roads that facilitate navigation, while minimizing extraneous details (see an example in Figure 12). We follow the approach of recent work on automated design of maps [Agrawala and Stolte 2001; Grabler et al. 2008] and analyze prior work in cognitive psychology [MacEachren 1995; Golledge 1999], cartographic generalization [Brassel and Weibel 1988; Battenfield and McMaster 1991; Harrie 2001] and high quality hand-designed destination maps, to better understand which roads are most important for a destination map and to develop a set of design principles for emphasizing those roads. Here, we present the principles we use in our system.

(P1) Hierarchical navigation. Cognitive psychologists have shown that people perceive and remember space hierarchically [Stevens and Coupe 1978; Tversky 1981]. For navigation tasks, the larger and faster highways are more salient than arterial roads, which in turn are more salient than residential streets. Indeed, drivers often plan routes hierarchically, first selecting the highways, then the arterials, and finally the residential streets, progressively increasing the level of detail as they get closer to the destination [Chase 1983; Car and Frank 1993]. Drivers familiar with a region often know how to navigate from their origin to the nearby highways and typically do not need the arterial and residential detail near the origin [Patel et al. 2006]. The most effective destination maps are designed to convey this hierarchical road structure and thereby facilitate navigation.

(P2) Complete traversable routes. Destination maps must depict complete and traversable routes to and from the destination. Two common heuristics for choosing individual routes are to select the fastest roads in order to reduce travel time and to minimize the number of turns in order to reduce complexity [Winter 2002]. Drivers

often choose routes that allow them to stay on the largest highways for the longest period of time – even at the expense of increasing the distance traveled. The set of routes should provide good spatial coverage over the entire region so that users from all surrounding directions can easily access the destination. However, the map should not be over-cluttered with irrelevant local roads far away from the destination.

(P3) Intersections as decision points. In navigation tasks, intersections are critical decision points where the driver must decide which road to follow next [Denis 1997; Allen 2000; Casakin et al. 2000]. Hand-designed destination maps concentrate information around intersections [Tversky and Lee 1999]. They provide the name of each road at the intersection so that drivers can find the corresponding road in the physical world. They also preserve the ordering of the roads around the intersection so that left turns remain left turns and right turns remain right turns. Similarly roads that continue straight through an intersection appear straight in the map.

(P4) Geometry and topology of roads. After turning onto a road the constraints imposed by the physical world make it easy for drivers to follow the road until they reach the next intersection. Thus, depicting the exact geometric shape of roads is far less important than accurately depicting the topology of the road network [Tversky and Lee 1999; Barkowsky et al. 2000; Agrawala and Stolte 2001]. Hand-designed maps commonly distort the road geometry but carefully preserve road topology. Mapmakers simplify the geometry of roads to straight lines or simple curves to reduce clutter, they reorient roads to visually separate roads that intersect at shallow angles and they distort the lengths of roads to remove extraneous information and further emphasize the intersections. However, to prevent over-distorting the geometry, mapmakers also preserve the overall shape of the road network and the headings of the roads (e.g. a north-south road remains oriented north-south), and they maintain the relative ordering of the roads by length so that short roads appear shorter than longer roads in the map. Because intersections are crucial decision points, and topological distortions such as false intersection can be extremely confusing, mapmakers ensure that such topological distortions never appear in hand-designed maps.

(P5) Simplification of highways and interchanges. Although people think of a highway as a single entity, most highways are divided into opposite sets of lanes that run parallel to one another, but may be separated by a relatively large distance. To reduce clutter, mapmakers often merge such divided highways and draw them as a single linear entity. Similarly, highway interchanges can be extremely complex with roads passing above, below and around one another as well as multiple on- and off-ramps between the highways and the surface roads. Accurately rendering such 3D interchanges in a 2D map can be difficult and therefore mapmakers often often simplify these complex interchanges into a single intersection point and either remove the ramps or draw them abstractly as simple curves.

3 Related Work

Our approach for automatically generating destination maps builds on several areas of previous work in automated mapmaking.

3.1 Road Selection

General-purpose online maps usually apply a scale-based selection filter to the road network. When the area of interest is relatively large (e.g. an entire state), the map shows only the largest-scale highways; as the user zooms in, the map progressively fades in the arterial roads and the residential streets. Some researchers have proposed more sophisticated techniques that analyze the connec-

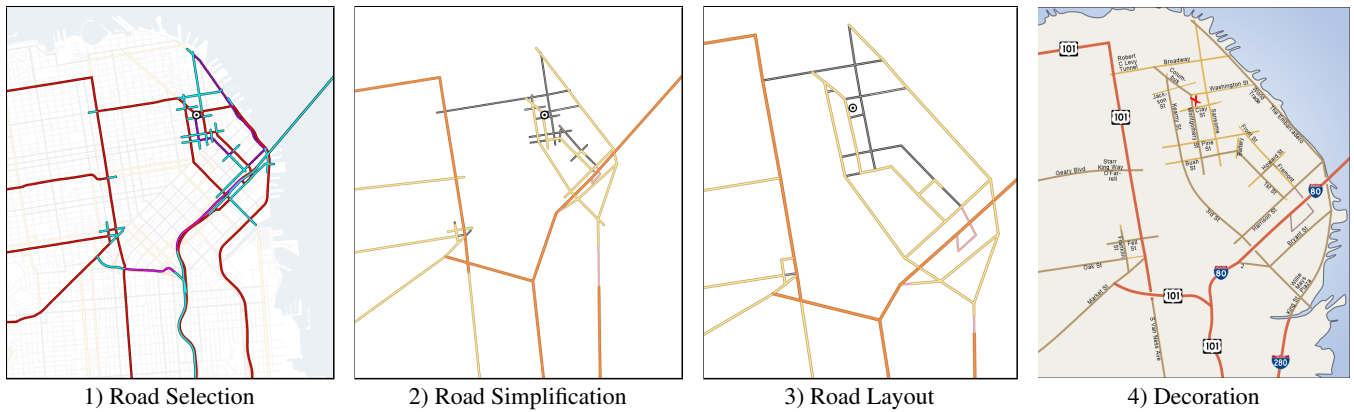


Figure 2: System overview. The input includes a destination and an area of interest. Our system is composed of 4 steps. 1) **Road selection** chooses the roads to include in the map (the color code is explained in Figure 3). 2) **Road simplification** merges the divided highways and complex highway interchanges. It also straightens the geometric shape of the roads. 3) **Road layout** computes the length, position and orientation of the roads such that all roads are visible. 4) **Decoration** adds labels and geographic context such as bodies of water to the map.

tivity structure of the road network [Mackaness and Beard 1993; Thomson and Richardson 1995; Jiang and Claramunt 2004] and semantic features such as traffic patterns [Grabler et al. 2008] to automatically infer the importance of roads. Bast *et al.* [Bast et al. 2007] leverage the intuition that long distance routes most often consist of using a small number of larger roads for the central section of the route. However, because these techniques are designed to aid navigation to any point within the area of interest rather than a single destination, they often include too much detail.

3.2 Map Layout and Rendering

Laying out a road network is a form of graph drawing [Di Battista et al. 1998]. Most graph drawing algorithms only focus on generating an intersection-free planar layout that preserves the topology of the graph. The constraints on laying out a network of roads are more stringent. Researchers have applied optimization techniques [Avelar and Müller 2000; Stott and Rodgers 2004] and direct geometric algorithms [Cabello et al. 2001] to schematize road networks (or networks of bus or subway lines). Wolff [2009] presents a comprehensive survey of these techniques. However, none of these methods directly rescale the roads and therefore they often fail to improve the visibility of short roads. While Merrick and Gudmundsson [2006] address the problem of rescaling roads based on network centrality, their approach does not include the geometric and topological constraints necessary to prevent over-distortion.

The primary problem with general-purpose online maps is that they use a fixed scale factor and therefore important roads are not visible. LineDrive [Agrawala and Stolte 2001] was the first system designed to directly solve the problem of rescaling roads to provide good visibility for all turning points in a route map. It uses a non-linear optimization procedure to lay out the roads and much like a hand sketched map, short roads are scaled up in length relative to the longer roads. While our work is inspired by LineDrive, there are many key differences that make the automatic design of destination maps significantly more difficult. First, LineDrive performs no road selection since it assumes a single point-to-point route given as input. In contrast, destination maps require careful selection of a road network that covers all likely routes from any origin in the surrounding area. Second, the layout optimization procedure in LineDrive operates a single linear sequence of road segments that form a simple polyline from the origin to the destination. Destination maps must lay out a 2D road network that contains multiple cy-

cles, which requires designing significant new objective functions and optimizations procedures to efficiently explore the layout design space. Towards this end, we discuss an energy function which when minimized robustly leads to maps that match a set of design principles. We develop both the form of the function as well as determine a set of empirically determined function parameters. Finally, we apply a perturbation-based optimization to minimize the energy function and define a new perturbation strategy that splits the road network around a single node and re-scale the roads on one side of the split.

An alternative to optimizing the layout of the road network graph is to simply apply image warping techniques to enlarge the most important regions of a fixed-scale, general-purpose map [Carpendale et al. 1995; Keahey and Robertson 1996; Böttger et al. 2008]. However, these methods often distort the shapes of roads in ways that can appear extremely unnatural. The distortions can make it difficult for users to mentally match the map with the physical world. Our approach is to combine optimization-based layout of the road network using a cost function that prevents such extreme distortion, with 2D image warping to add in geographic area landmarks.

4 Generating Destination Maps

To create a map with our system, a user specifies a destination, either by clicking on a general-purpose map or by typing in an address. The user also specifies a rectangular area of interest surrounding the destination. We set the output aspect ratio to 1:1, 2:3, or 3:2, whichever is closest to the input aspect ratio. The output size is set so that the resulting map fits onto a letter-sized sheet of paper.

A database stores the complete network of roads in North America. Edges represent straight road segments and nodes represent either bends within a road or road intersections. Each edge stores the name(s) of the road, maximum speed, and one of six functional classes: highway, major road, arterial, street, ramp, or ferry line. The database also includes a vector representation of all bodies of water. As shown in Figure 2, our system generates a destination map via a sequence of four steps; 1) Road selection, 2) Road simplification, 3) Road layout and 4) Decoration.

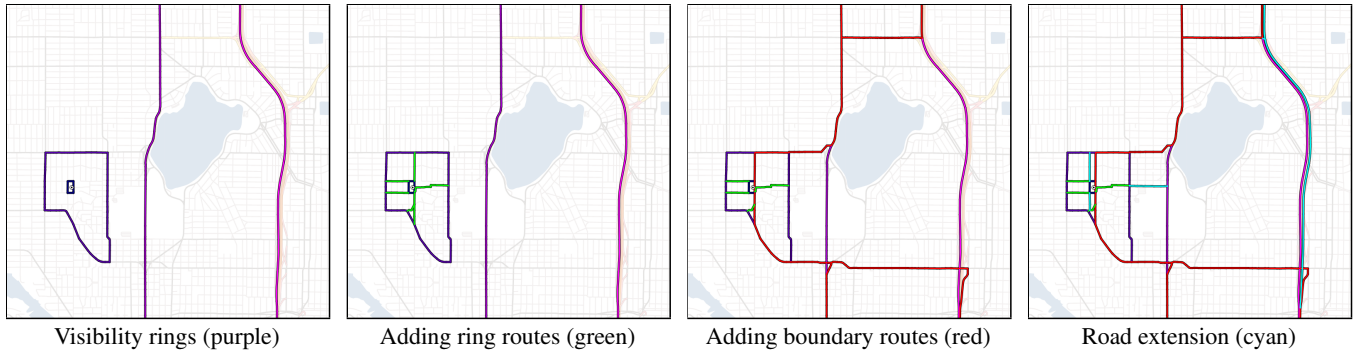


Figure 3: Road selection. Our system hierarchically computes **visibility rings** for each class of roads (purple). It then generates complete **traversable routes** from the nodes on the rings (green) and the boundary (red) of the area of interest to the destination. Finally the **road extension** step adds additional road segments for context (cyan).

4.1 Road Selection

Roads are selected based on the design principles of (P1) hierarchical navigation and (P2) producing complete traversable routes to the destination from anywhere in the surrounding area. Road selection proceeds in a series of three steps, as shown in Figure 3:

- **Visibility rings:** We first compute concentric rings of highways, arterials and residential streets around each destination. These visibility rings form the hierarchy associated with navigating to a destination.
- **Traversable routes:** To produce complete traversable routes, we use a shortest path algorithm to connect the rings to the destination. We also connect all highways entering the boundary of the area of interest to the destination.
- **Road extension:** Some selected road segments are extended to provide additional context. For example, if two disjoint segments of the same road have been selected, we add the segments in between.

A common procedure in our road selection algorithm is to compute a path between a given pair of nodes. Such point-to-point route planning is a well studied problem that is often solved using Dijkstra’s shortest path algorithm or A^* search [Sanders and Schultes 2007]. To produce fast and simple routes, we follow Winter’s [2002] suggestion and use a cost function that weights each road segment based on the time required to traverse it at maximum speed and add a fixed penalty (10 seconds in our implementation) for taking turns. This cost function produces fast routes that favor highways while minimizing the number of turns to maintain simplicity.

4.1.1 Visibility Rings

Mimicking the hierarchical navigation (P1) drivers commonly use in route planning, we compute a nested set of visibility rings around the destination (Figure 3 leftmost panel). Starting with the highways we use a 2D radial visibility sweep algorithm [de Berg et al. 2008] to identify the ring of highways visible from the destination. We similarly compute visibility rings for each smaller road class; the major roads, arterials, and streets. At each stage, roads belonging to larger classes block the visibility of smaller class roads.

If the rings are non-convex, the radial sweep returns only the visible portions of the ring and the loop remains unclosed. Such open rings also occur when there are no roads of the ring’s class in some directions. We attempt to close such openings using the shortest path

algorithm between the open endpoints. To avoid clutter, we only close the ring if the closure is no more than twice the length of the straight line connecting the open endpoints.

4.1.2 Traversable Routes

To add local context around the destination, we connect the street and arterial visibility rings to the destination with complete traversable routes (P2). We use Dijkstra’s algorithm to compute the shortest path from every ring node to the destination (Figure 3 second panel). To avoid cluttering the map with too many paths we reduce the cost of ring roads by 70% to favor traveling on these edges.

We also add routes from all important roads entering the area of interest (Figure 3 third panel). We sort the list of all roads entering the area of interest based on importance criteria including their functional class, whether they are part of a national highway system (e.g. US interstates or state routes), and maximum speed. We then iteratively add a traversable route from a point at which that road crosses the area of interest boundary to the destination. When computing these routes, we discount traveling on visibility rings by 30%. We use a less aggressive discount than for the visibility ring connections because we want to ensure that the routes from the boundary are close to optimal. After each iteration, we remove all roads from the list whose boundary crossing is closer than some threshold to the just selected road.

4.1.3 Road Extension

The visibility rings and traversable routes provide a minimal road network for a destination map. To add further context, we extend each set of contiguous roads segments selected in earlier steps by following them outwards from both endpoints and adding any unselected edges until the road name changes (Figure 4 left). We extend highways to the edge of the area of interest, but smaller roads only up to a maximum of 1.5 km.

Finally, we eliminate all dangling dead-end road segments. Specifically, we recursively remove edges that are connected to a node with valence equal to one (Figure 4). As an exception, we never remove highway edges and edges belonging to traversable routes, as these are important for navigating the map.

Such dead-end elimination decreases the number of edges in the network and therefore increases flexibility for our road layout algorithm. However, when removing dead-ends, we keep short extensions, or *tails*, to provide more context at intersections (Figure 4 right), showing whether it is an X-crossing or T-junction.

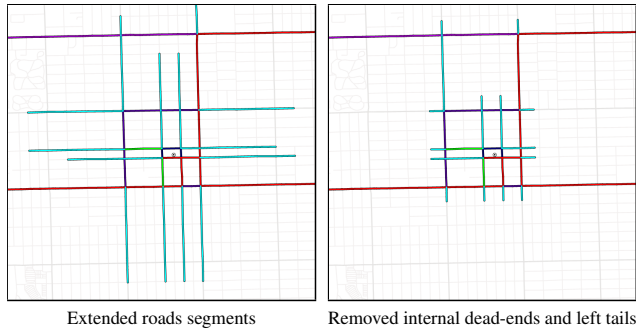


Figure 4: Road extension. Left: roads extended up to 1.5 kilometers. Right: removed dangling dead-ends, but kept short tails to indicate the intersection types. This figure uses the same color coding as Figure 3.

4.2 Road Simplification

The full geometric complexity of roads is not necessary for navigation and clutters maps with irrelevant detail. Our simplification step is designed to significantly reduce geometric detail based on design principles (P4), which recommends distorting road geometry to straight lines or simple curves while preserving topology and (P5), which recommends simplifying divided highways and interchanges. Since simplification also significantly speeds up the road layout algorithm described in the next section, we aggressively simplify roads in this stage, and then re-introduce some of the detail after road layout. The simplification algorithms are applied to the graph that contains the subset of selected roads.

4.2.1 Merging Divided Roads and Simplifying Interchanges

The first step in road simplification is to merge divided roads, e.g. most highways. We first mark all nodes with valence not equal to two, as these nodes represent places where a road begins or ends, a lane splits, or a ramp peels off (marked with red dots in Figure 5). We also mark all nodes where a road name changes occur.

Then, for each marked node, we spatially search for parallel lanes with the same road name within a radius of 200 meters. For each such parallel lane, we insert a corresponding node at the nearest point (marked with green dots in Figure 5). Corresponding nodes are merged into a single node. If two merged nodes are now connected by more than one sequence of edges, we collapse them into a single sequence by retaining the one of the highest functional class, or in case of a tie, the shortest sequence.

Highway interchanges often form a complex tangle of ramps connecting two highways, or a highway to a surface road. We remove ramps to simplify such interchanges and convert the overpass or underpass into a simple intersection node. Two common cases of such ramp removal are shown in Figure 6.

4.2.2 Geometry Simplification

To simplify the geometric shape of roads, we apply a subtractive variant of the Ramer-Douglas-Peucker algorithm [Ramer 1972; Douglas and Peucker 1973], modified to preserve the topology of the road network. First we fix all nodes with valence other than two, removing them from consideration in the simplification process. Valence one nodes represent road endpoints, usually at the boundary of the area of interest while valences greater than two represent an intersection between two or more roads. We fix such endpoints and intersection nodes to ensure that the connectivity of

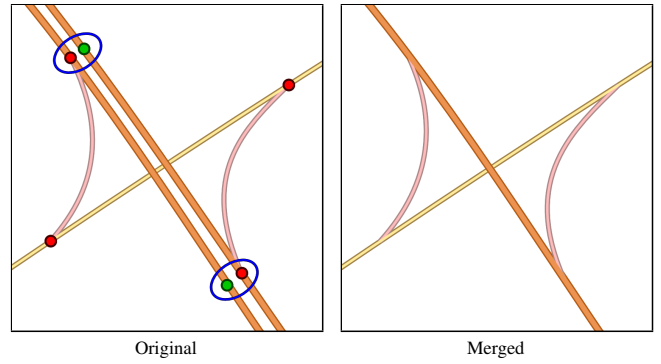


Figure 5: Merging divided highways. Left: Red dots mark nodes with valence unequal to two. Green dots mark the closest points on parallel lanes. Grouped nodes are encircled. Right: Result after collapsing grouped nodes.

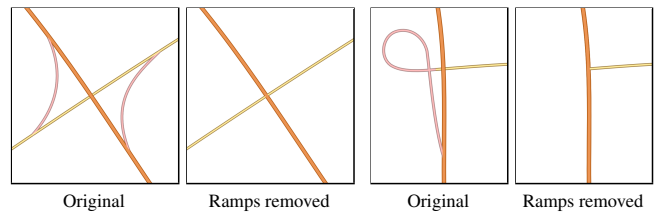


Figure 6: Interchange simplification. We remove the ramps (pink) and convert the overpasses into regular junctions to preserve proper connectivity.

the road network is preserved. The Ramer-Douglas-Peucker algorithm then iteratively removes the *unfixed* nodes in order of their importance, where importance is proportional to the deviation of the node from the line connecting the two neighboring nodes.

To preserve sharp bends, we only remove a node that forms an angle smaller than 130° near the destination, reduced to less than 95° at the boundary of the area of interest. The varying threshold preserves more geometric details of the local roads around the destination, which usually expands more in the final destination map than bigger peripheral roads, which do not expand or may even contract.

We also reject a node removal if it would change the topology of the road network by introducing a false intersection. We continue removing nodes in this greedy manner until no further removal is possible.

4.3 Road Layout

The goal of the road layout step is to ensure that all of the roads in the selected road network are visible. Based on design principles (P3) and (P4), the layout algorithm has flexibility to adjust the length, position, and orientation of each road in the network, but it must preserve network topology, and it should preserve the overall road headings and the shape of the network.

To lay out the network, we modify the nonlinear optimization strategy of LineDrive [Agrawala and Stolte 2001]. Starting from the initial fixed-scale layout, we define a *cost function* that evaluates the quality of the layout and a set of *moves* that adjust some property of the current layout to produce a new layout. The moves define a space of possible layouts, and the optimizer finds the lowest cost layout in this space.

The simplest such search-based optimization strategy is to define an objective function and to then only accept moves that decrease the cost and reject all others. However, since our cost function may include local minima, we adopt a nonlinear optimization strategy that can avoid such minima. While LineDrive used simulated annealing, we obtain better results using an optimization technique known as a *continuation* method [Allgower and Georg 1990]. Simulated annealing includes a probability (gradually reduced by a cooling schedule) of accepting a move that increases cost. In contrast, continuation methods only accepts moves that decrease the cost. However, the cost function itself is modified as the optimization proceeds, beginning with a smoother function and proceeding to the final cost function.

We first describe how we precompute properties of the selected road network required for the optimization. We then describe the cost function and the layout moves.

4.3.1 Precomputation

To increase the efficiency of the optimization we precompute the *relative orientation* relationships between pairs of edges. We mark connected edges as *straight* if their orientations differ by less than 5° . We mark two connected edges as mutually *orthogonal* if their orientations differ by 85° to 95° . Similarly, we mark two non-connected segments as mutually *parallel* if their orientations differ by less than 5° and they are *directly visible* to one another (i.e., a line perpendicular to one of them intersects the other and does not cross any other edge). The cost function in the following section attempts to preserve these relative orientations wherever possible.

4.3.2 Cost Function

Our cost function was developed through extensive empirical study. It is designed to push the optimization towards a layout in which roads of all classes are visible, but that also preserves the topology of the network as well as the overall headings of the roads in the network. The cost of a layout is a weighted sum of five terms.

$$E = \frac{\lambda_{int}e_{int} + \lambda_{len}e_{len} + \lambda_{rlen}e_{rlen}}{\lambda_{ang}e_{ang} + \lambda_{rang}e_{rang}} \quad (1)$$

where the e 's represent the terms and the λ 's represent the weights. The individual cost terms include:

1. **False intersections:** To prevent changes to the network topology, false intersections incur an infinitely large cost. Moreover, two edges may *appear* to intersect if the distance between them is very small and such perceived intersections can be very confusing. Thus, we penalize edges that lie closer than $d_{int} = 8\text{mm}$ to one another:

$$e_{int} = \begin{cases} \infty, & \text{if false intersection} \\ \sum_i \left(\frac{d_{int} - \min(d_i, d_{int})}{d_{int}} \right)^2, & \text{else} \end{cases} \quad (2)$$

where the summation is over the edges and d_i is the shortest distance between edge i and any other edge it is not directly connected to.

2. **Minimum length:** Every edge should maintain a minimum size on the screen so that viewers can easily see the extent of the road segment and turning points or intersections along it. The minimum length should also be long enough so that the road can be labeled easily. Thus, we penalize edges whose length l_i is shorter than $d_{len} = 2.5\text{cm}$:

$$e_{len} = \sum_i (d_{len} - \min(d_{len}, l_i))^2 \quad (3)$$

3. **Relative length:** While satisfying the minimum length requirement, it is also preferable to maintain the relative length between segments, we therefore define:

$$\begin{aligned} e_{rlen} &= \sum_{i,j} \left(\frac{r_0(i,j) - r(i,j)}{r(i,j)} \right)^2, \\ r_0(i,j) &= \begin{cases} l_{i_0}/l_{j_0}, & \text{if } l_{i_0} > l_{j_0}, \\ l_{j_0}/l_{i_0}, & \text{else} \end{cases} \\ r(i,j) &= \begin{cases} l_i/l_j, & \text{if } l_{i_0} > l_{j_0}, \\ l_j/l_i, & \text{else} \end{cases} \end{aligned} \quad (4)$$

where l_{i_0} and l_{j_0} represent the original lengths of segments i and j , while l_i and l_j represent their current lengths.

4. **Orientation:** Edges should retain their original orientation. The cost is the square of the angular deviation (in radians), $\Delta\theta_i$, from their original orientation:

$$e_{ang} = \sum_i \Delta\theta_i^2. \quad (5)$$

5. **Relative orientations:** Segments marked as parallel, straight, or orthogonal should keep their relative orientations. Thus, we penalize deviations from their perfect relative orientation:

$$e_{rang} = \sum_{(i,j) \in S_{\parallel}} \Delta\phi_{i,j}^2 + \sum_{(i,j) \in S_{\perp}} \Delta\phi_{i,j}^2 + \sum_{(i,j) \in S_{\perp}} (\Delta\phi_{i,j} - \pi/2)^2, \quad (6)$$

where $\Delta\phi_{i,j}$ is the relative orientation (in radians) of segments i and j , and S_{\parallel} , S_{\perp} , and S_{\perp} are the sets of pairs of segments marked as parallel, straight, and orthogonal, respectively.

4.3.3 Layout Perturbation

The cost function contains a large number of variables (the node locations) and includes both nonlinearities and discontinuities (e.g., e_{int}). Thus, computing derivatives for optimization via gradient descent is difficult. Instead, we apply a direct search method that perturbs the layout using discrete stochastic *moves* that push the network towards a lower cost layout.

The optimizer applies a layout move on each iteration, computes the cost of the new layout, and accepts it if it reduces cost. When a move is accepted, we uniformly rescale the positions of all the nodes so that the network just fits within the boundary of the map. We also snap nodes that originally lay on the boundary back to the boundary.

The individual moves can take on many forms. In developing the optimization method we sought moves that would tend to preserve topology and minimize distortion while allowing enough flexibility to efficiently minimize the objective. After extensive testing, we settled on a move in which we pick a node, then choose

1. an arbitrary line through the node,
2. a side,
3. a scale factor in $[0.95, 1.05]$, and
4. either radial or orthogonal.

Then, scale the chosen side of the layout by the scale factor either outward for scale factors greater than 1 or inward for smaller scale factors. The scaling is performed either radially outward from the node or orthogonal to the line, depending on the last stochastic choice above.

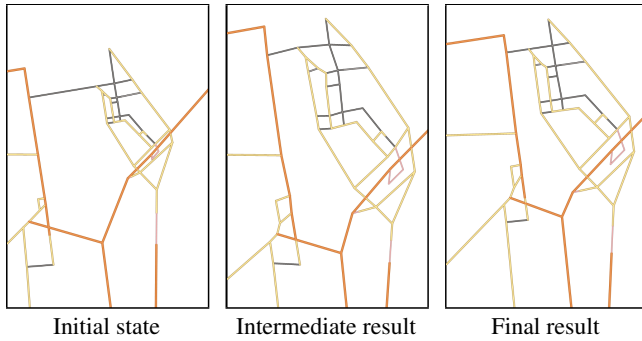


Figure 7: Different states of the optimization.

4.3.4 Optimization

The total cost of any proposed layout is a weighted sum of the energy terms defined in Section 4.3.2. The continuation method for the optimization (Section 4.3), involves setting the weights, i.e., the λ 's in Equation (1), in two stages. Initially, we use a smoother cost function with fewer terms and then add more terms in the second stage.

In the first stage, we heavily weight the minimum length requirement, while reducing the influence of the orientation related terms. Intuitively, this weighting helps remove the local minima near the initial fixed-scale layout by allowing the local regions around the destinations to grow without having to maintain road orientations perfectly. In the second stage, the orientation terms are increased, leading to a more aligned final result in which the heading of the road segments better reflects their original headings.

More specifically, we set the weights at each stage as follows:

Term	λ_{int}	λ_{len}	λ_{rlen}	λ_{ang}	λ_{rang}
Stage 1	100	1000	1	1000	0
Stage 2	100	1000	1	5000	500

During the optimization, we check the energy every 500 moves and terminate if it has decreased by less than 1% since the last check. An example of an initial layout and the layout at the end of each stage is shown in Figure 7.

4.4 Decoration

The final step before the actual rendering of the destination map is to add back in geometric road details, road labels, and geographic features.

We reintroduce some geometric detail removed during the simplification stage. We are careful not to introduce any false or perceived intersections. We begin with every road segment being a straight line, and then gradually reintroduce detail by linearly interpolating between the straight line and the detailed geometry. Interpolation weights are set to insure that perceived false intersections never occur.

The background is drawn either as a solid color or a paper texture. We have implemented a number of styles for drawing the roads, as can be seen in Figure 15. For a traditional looking map, we render the highways widest and in orange, arterials in yellow, and residential streets in white. The widths depend on whether *European* style is chosen, which allows sufficient width to insert one row of text labels within the road, or an *American* style, which results in thinner roads for which all text is placed outside the road. A more informal

	Road Selection	Simplification	Layout	Decoration	Total
Median	1.82s	1.43s	3.77s	1.84s	10.06s
Average	3.54s	3.08s	4.48s	2.82s	13.92s
Standard Deviation	6.89s	4.48s	3.56s	3.19s	13.44s
20th Percentile	0.49s	0.34s	1.88s	0.85s	4.48s
80th Percentile	3.09s	3.74s	6.23s	3.92s	17.80s

Figure 8: Timing results.

sketchy look of hand drawn maps is created by repeatedly over-drawing noisy lines: ten times for highways, five times for major roads, three times for arterials, and once for smaller streets.

The database includes the outlines of bodies of water (other geographic entities could be handles in a similar fashion). Our goal is to warp the water features to correspond to the repositioning of the roads in the layout stage. We seek a free form deformation of the original map that aligns the original road positions with the new ones and does not incur any fold overs. We leverage the work of Lee *et al.* [1995], which performs a multi-level FFD based on B-splines to construct a C^2 deformation. The constraints consist of the pre- and after-layout locations of the roads. Given this deformation, we warp the vertices of the original water features to the new layout. Four styles to render the water can be seen in Figure 15.

4.4.1 Label Placement

Automatic label placement is a well-studied problem, particularly in the context of automated map design [Christensen *et al.* 1995; Wolff 2009]. We do not add to the extensive literature in this work, but rather extend the multi-criteria optimization strategy of Agrawala and Stolte [2001] to label our roads.

Our layout criteria are designed to label each road at least once; for long roads such as highways, we attempt to place one label every 8cm. Our road database often contains several names for the same stretch of road. For example state highways may have a numerical name such as CA-101 as well as local names such as Bayshore Freeway or the James Lick Freeway. If roads are part of a national highway system (e.g. US state routes), we label them with highway shield symbols and ignore all text names to reduce clutter. For smaller roads, we try to place a label for every name if space permits.

4.5 Performance

Destination Maps is implemented in C# using the Silverlight framework. The application runs in a web browser on the user's machine. The road network data is downloaded from an online database. Figure 8 reports timings for the maps in our test batch, broken down to the individual algorithm stages. We did not include download time, as it might vary depending on the user's connection speed. On a regular broad-band connection, the download time takes about 5–15 seconds. This time is typically hidden from the user, since we begin downloading while they define the region of interest.

While the system does not operate in real-time, it is fast enough for a web application aimed at producing maps for invitations, flyers, business cards, etc.

5 Evaluation

We have evaluated the destination maps created with our system in a number of ways. We first numerically evaluated the need for each objective term and how well the each objective is met. A formal user study was conducted with 45 volunteer subjects to assess the

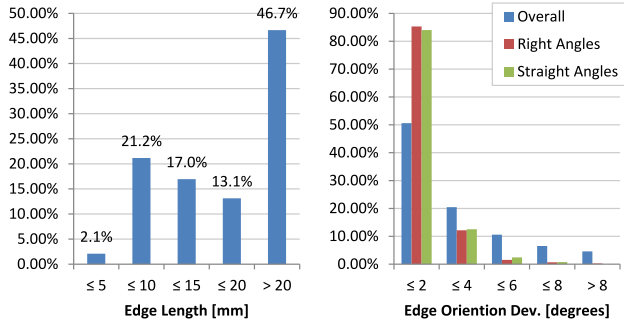


Figure 9: Left: Percentage of edge lengths that fall below a minimum length objective. Right: Percentage of edges that deviate from the objective by degrees.

utility of the destination maps. We also released Destination Maps as an application resulting in approximately 57,000 maps created so far. Approximately 1% of users filled out an optional feedback page which we report on. We also made an informal comparison to Hand Drawn Maps. We report on each of these methodologies in turn.

5.1 Analyzing the Objectives

We created a test set of 72 locations including a wide range of different locations to analyze the performance of the destination map system. The maps are shown in the supplementary materials, and in figures throughout the paper. To analyze the effectiveness of each term in the objective function, we selectively turned off the energy terms one by one. By manual inspection, we found that leaving out any term resulted in artifacts in some maps. We show a few examples in the supplementary materials.

Figure 9 shows histograms of edge lengths and edge orientations that deviate from the objectives.

We also recorded violations of the design principles. The hard constraints were always satisfied: there are no missing or false intersections, no interior/boundary node left/entered the interior of the map. 90% of all edges have are almost perfectly oriented and clearly visible in the resulting map with lengths greater than 10mm. 90% of all right/straight angles have a deviation of less than 2.5° respectively. Design principles enforced using soft constraints are occasionally violated. 0.5% of all edges came closer than 2mm to other edges. Very rarely a turn's orientation gets flipped (0.35% of the turns).

5.2 Formal User Study

We also carried out a formal user study to assess the perceived usability of the resulting destination maps vs. fixed scale maps. 45 volunteer subjects, all experienced computer users, 12 female and 33 male, were asked to supply a US address they were familiar with outside the local area. The addresses covered a wide range both geographically as well as ranging across urban, suburban, and rural locations. We prepared destination maps and corresponding fixed scale maps with the same area of interest for each address.

Each subject was tested on three locations; two destinations they were not familiar with (drawn from the others' familiar addresses) and their own designated familiar destination. They were presented with either the fixed or destination map first, and then the other. For each map they were asked: "Assuming you were familiar with the largest roads depicted, how comfortable would you be navigating

		Destination Maps					Σ
		Very Uncomfortable				Very Comfortable	
Fixed Scale Map	Very Uncomfortable	1	9	0	15	2	27
		8	7	2	21	7	45
		0	5	1	2	2	10
		1	3	4	11	13	32
	Very Comfortable	3	1	0	9	8	21
	Σ	13	25	7	58	32	

Figure 10: Histogram of ratings for comfort levels navigating with fixed scale map vs. destination map for the same destination.

to the destination with only this map?" A 5-point Likert scale (very uncomfortable to very comfortable) was offered for the responses. For the familiar location, the wording was slightly different: "How comfortable would you be giving this map to a friend to navigate to the destination with only this map?" Both maps were then presented side-by-side and the users were asked which map they would prefer to navigate to the destination. Again a 5-point scale was used (greatly prefer fixed scale map to greatly prefer destination map).

To gather more general feedback, three open ended questions asked: "What would be your biggest concern with having only this map with you?" for each of the maps, and "What would you add or remove from this map to make it more useful?" for the destination maps only.

Results: Overall, the comfort level and preference results indicate a significant leaning towards destination maps over the fixed scale maps. The table in Figure 10 shows the number of paired maps (fixed,destination) of each location with particular comfort rating pairings. The graphs in Figure 11 show the summary "comfort" and "preference" results, broken down by unfamiliar and familiar locations. Across all 45 subjects for all 3 destinations (90 unfamiliar + 45 familiar = 135 total pairs of maps), 53% of the fixed scale maps elicited either very uncomfortable or uncomfortable ratings, while 39% indicated being very comfortable or comfortable with them, with 8% neutral. In contrast, only 28% of the destination maps were rated as very uncomfortable or comfortable to navigate with alone, while 67% were rated as very comfortable or comfortable as navigation aids, with 7 (5%) neutral. The side-by-side preferences echoed this trend with 28% of the fixed maps preferred over the equivalent destination map while 65% of the destination maps were preferred, with 9 a toss-up. A chi-squared analysis comparing the 5x5 table of comfort responses (Figure 10) to a null hypothesis of a uniform distribution yields a $p = 0.01$ that the observed ratings are random.

The open ended comments provided very valuable feedback as well. The most common perceived and specific problems with the destination maps categorized and ordered by their frequency were:

1. A concern that if one makes a mistake navigating with the destination map, it is hard to get back on track.
2. A lack of ability to understand the distances for each segment of the routes.
3. A desire to have landmarks shown.
4. A desire to see exit numbers from highways.
5. Some problems with road labeling.
6. A desire to see more cross streets or more streets in general for context.
7. In the familiar destinations, favored routes not depicted.

This will be invaluable as we continue to improve the maps.

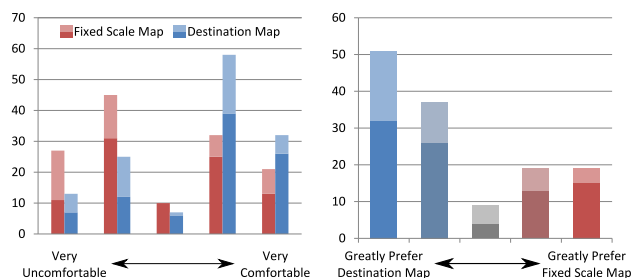


Figure 11: Left: Total number of ratings for comfort levels navigating with fixed scale map only (red) vs. destination maps only (blue). Light shaded regions are the portion for familiar locations and the dark shaded regions correspond to the unfamiliar location totals. Right: Number of preference ratings reported if they could only take one map: fixed maps (red) and destination maps (blue).

5.3 Informal User Study

We recently placed Destination Maps online as a Bing Maps App for anyone to try out (see Destination Maps under *Map Apps* at <http://www.bing.com/maps/explore/>). As of the final draft, over 57,000 maps have been generated. Users indicate a destination and area of interest and less than 1 minute later are presented with a destination map. Styles can then be changed interactively. We examined the feedback from the first 19,000 destination maps generated through the online system. An optional feedback page was filled in by 211 users (about 1%). Feedback was given in the form of whether the user would feel comfortable giving this map to a friend, or would prefer a standard fixed scale map. An open text box was also provided for general feedback.

Of the 211 responses, approximately 55% indicated a positive reaction either through indicating a comfort/preference for the destination map and/or through the general feedback. Many responses were very complimentary. Approximately 37% said they would prefer the fixed map and/or registered specific problems with the map. The most common complaints were lack of detail, and/or missing roads. Some indicated a desire to annotate the maps or to be able to have more than 1 destination. The remaining 8% either left the feedback blank or provided indecisive remarks.

5.4 Comparison to Hand Drawn Destination Maps

In Figure 12, we compare our results to a professionally designed destination map found on the internet. The overall selection of roads and layout is similar. However, the hand designed maps make much more creative use of annotations.

6 Discussion and Future Work

We have presented a system for creating destination maps to provide users a means to navigate to a given location from anywhere in a given area of interest. Creation of the maps involves selecting appropriate roads, laying them out in a 2D rectangle and rendering them in a number of styles.

We have quantitatively evaluated 72 destination maps depicting random locations based on their adherence to the stated objectives. We also qualitatively evaluated 45 maps through a formal user study, and 211 non-randomly sampled maps through an optional feedback form in an online system used to generate over 50,000 destination maps so far. The fact that a significant majority of users indicated

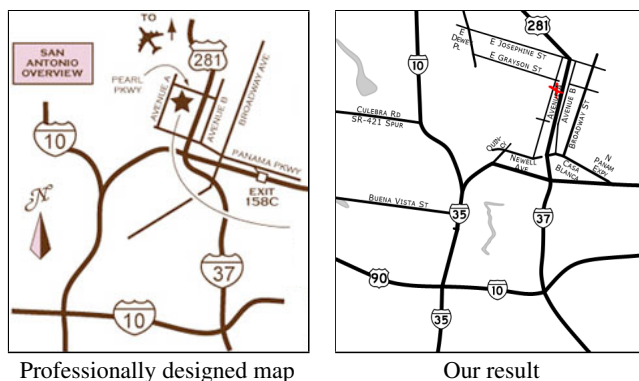


Figure 12: Comparison of professionally designed destination map with our automatic result.

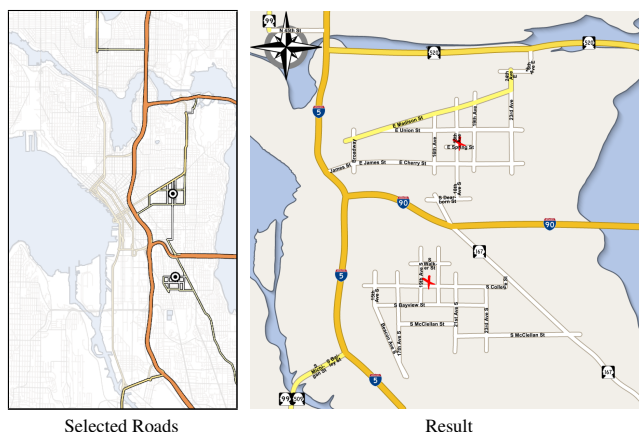


Figure 13: Map with two destinations.

a preference for the destination maps is a very strong result, especially given that the current alternative is only the fixed scale maps. The combination of having both fixed and destination maps would certainly raise both the comfort level and navigability over either individually.

The feedback raises a number of issues we hope to address in the next iteration of destination maps. For those familiar with the location, some of the maps do not depict a particular favored route. A shortest route algorithm is only as good as the underlying data. For example, the data does not contain high level knowledge such as that one roadway has lots of lights and traffic and another is, on average, faster. We are designing a future system that (1) allows some user editing to indicate a preferred route to improve the map, and (2) uses this editing session to update the data for future maps.

Many hand drawn destination maps contain multiple destinations, e.g., the locations of the church and the reception venue for a wedding map. We experimentally added this feature to our system. In this case, we added additional traversable routes in both directions between each pair of destinations in Section 4.1.2. Figure 13 shows an example of a map with two destinations.

Hand drawn maps also often include many textual and graphical annotations such as landmark icons or water body names. While landmarks are not essential for navigation, they do provide context that can help navigators match the map to the physical world. Several projects have been designed to automatically select and annotate city road maps with photos or renderings of landmarks [Raubal and

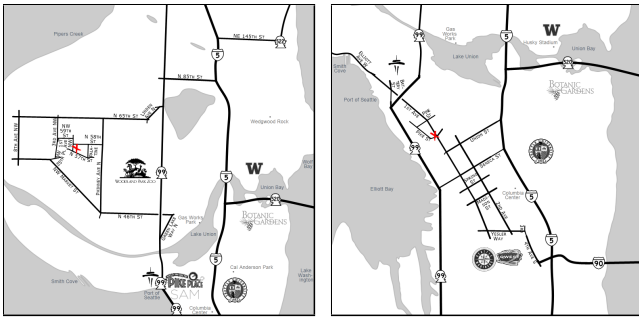


Figure 14: Automatically placed landmarks.

Winter 2002; Grabler et al. 2008] to generate tourist maps. We have begun to experiment with automated landmark placement as well. Figure 14 shows two maps generated by algorithmically placing a single importance ordered set of textual and iconic landmarks in Seattle. When sufficient space is found near the landmark's lat/long location, the text or icon are inserted. We also plan to add annotations for one-way streets and to indicate preferred routes from a particular direction. Test subjects also asked for segment length indicators which we hope to add as well.

Finally, one might apply the general ideas for network route visualization to other networks such as large molecular structures containing a hierarchy of links with varying importance.

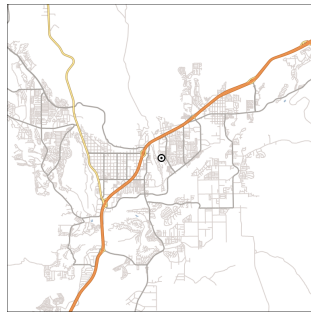
Future work aside, our system can already construct maps that are visually similar to hand-designed destination maps in terms of road selection and layout. We have outlined a comprehensive set of design principles for both selecting the roads and geometrically distorting their layout in a way that clarifies the location of the destination. These principles are echoed in a series of novel algorithms for road selection, simplification, and layout. We have demonstrated that these algorithms can automatically generate informative destination maps. Both formal and informal user studies indicate a preference for destination maps over fixed scale maps.

References

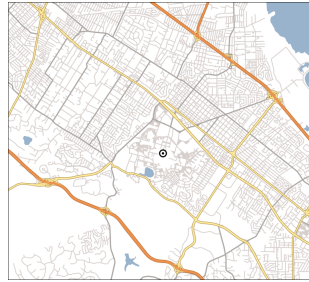
- AGRAWALA, M., AND STOLTE, C. 2001. Rendering effective route maps: improving usability through generalization. *Proceedings of SIGGRAPH 2001*, 241–249.
- ALLEN, G. L. 2000. Principles and practices for communicating route knowledge. *Applied Cognitive Psychology* 14, 4.
- ALLGOWER, E. L., AND GEORG, K. 1990. *Numerical continuation methods: an introduction*. Springer-Verlag New York, Inc., New York, NY, USA.
- AVELAR, S., AND MÜLLER, M. 2000. Generating topologically correct schematic maps. *Proc. 9th International Symposium on Spatial Data Handling*, 4–28.
- BARKOWSKY, T., LATECKI, L. J., AND RICHTER, K.-F. 2000. Schematizing maps: Simplification of geographic shape by discrete curve evolution. In *Spatial Cognition II*, Springer-Verlag, C. H. C. Freska, W. Brauer and K. Wender, Eds., 41–53.
- BAST, H., FUNKE, S., SANDERS, P., AND SCHULTES, D. 2007. Fast routing in road networks with transit nodes. *Science* 316, 5824 (April), 566.
- BÖTTGER, J., BRANDES, U., DEUSSEN, O., AND ZIEZOLD, H. 2008. Map warping for the annotation of metro maps. *Computer Graphics and Applications*, 56–65.
- BRASSEL, K., AND WEIBEL, R. 1988. A review and conceptual framework of automated map generalization. *International Journal of GIS* 2, 3, 229–244.
- BUTTENFIELD, B. P., AND MCMASTER, R. B., Eds. 1991. *Map Generalization: Making rules for knowledge representation*. Longman Scientific.
- CABELLO, S., DE BERG, M., VAN DIJK, S., VAN KREVELD, M., AND STRIJK, T. 2001. Schematization of road networks. In *Proc. Symp. on Comp. Geom.*, 33–39.
- CAR, A., AND FRANK, A. 1993. Hierarchical street networks as a conceptual model for efficient way finding. In *Proc. EGIS*, 134–139.
- CARPENDALE, M., COWPERTHWAIT, D., AND FRACCHIA, F. 1995. Three-dimensional pliable surfaces: For effective presentation of visual information. In *Proc. UIST*, 217–226.
- CASAKIN, H., BARKOWSKY, T., KLIPPEL, A., AND FREKSA, C. 2000. Schematic maps as wayfinding aids. *Lecture Notes in Computer Science* 1849, 54–71.
- CHASE, W. G. 1983. Spatial representations of taxi drivers. In *Acquisition of Symbolic Skills*, New York: Plenum Press, D. R. Rogers and J. A. Sloboda, Eds., 391–405.
- CHRISTENSEN, J., MARKS, J., AND SHIEBER, S. 1995. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics (TOG)* 14, 3, 203–232.
- DE BERG, M., CHEONG, O., VAN KREVELD, M., AND OVERMARS, M. 2008. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, March.
- DENIS, M. 1997. The description of routes: A cognitive approach to the production of spatial discourse. *Cahiers de Psychologie Cognitive* 16, 4, 409–458.
- DI BATTISTA, G., EADES, P., TAMASSIA, R., AND TOLLIS, I. 1998. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR Upper Saddle River, NJ, USA.
- DOUGLAS, D., AND PEUCKER, T. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer* 10, 2, 112–122.
- GOLLEDGE, R. 1999. *Wayfinding behavior: Cognitive mapping and other spatial processes*. Johns Hopkins Univ Press.
- GRABLER, F., AGRAWALA, M., SUMNER, R. W., AND PAULY, M. 2008. Automatic generation of tourist maps. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27, 3 (Aug.), 100:1–100:11.
- HARRIE, L. 2001. *An optimisation approach to cartographic generalisation*. Dept. of Technology and Society, Lund Institute of Technology, Lund University.
- JIANG, B., AND CLARAMUNT, C. 2004. Topological analysis of urban street networks. *Environment and Planning B* 31, 1, 151–162.
- KEAHEY, T., AND ROBERTSON, E. 1996. Techniques for non-linear magnification transformations. *Proc. IEEE Info. Vis.*, 38–45.
- LEE, S.-Y., CHWA, K.-Y., SHIN, S. Y., AND WOLBERG, G. 1995. Image metamorphosis using snakes and free-form deformations. *Proceedings of SIGGRAPH 1995*.
- MACEACHREN, A. M. 1995. *How Maps Work*. The Guilford Press.



Princeton Map



St. George Map



Stanford Map



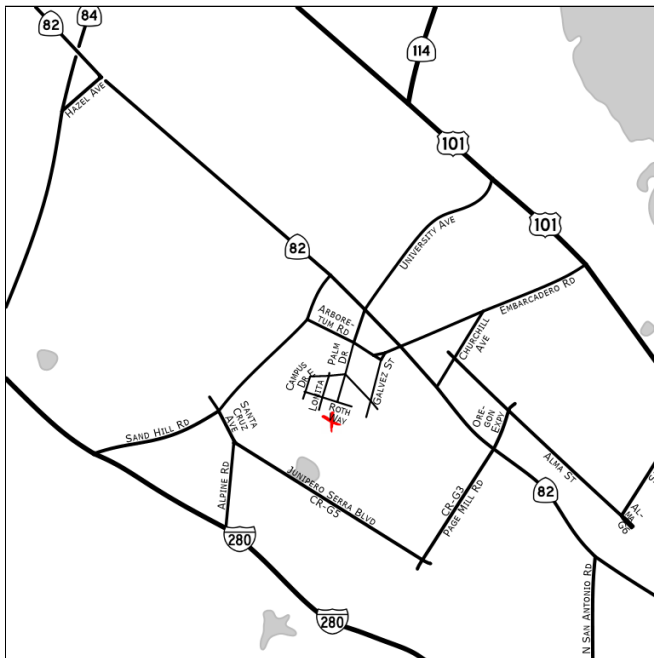
Seattle Map



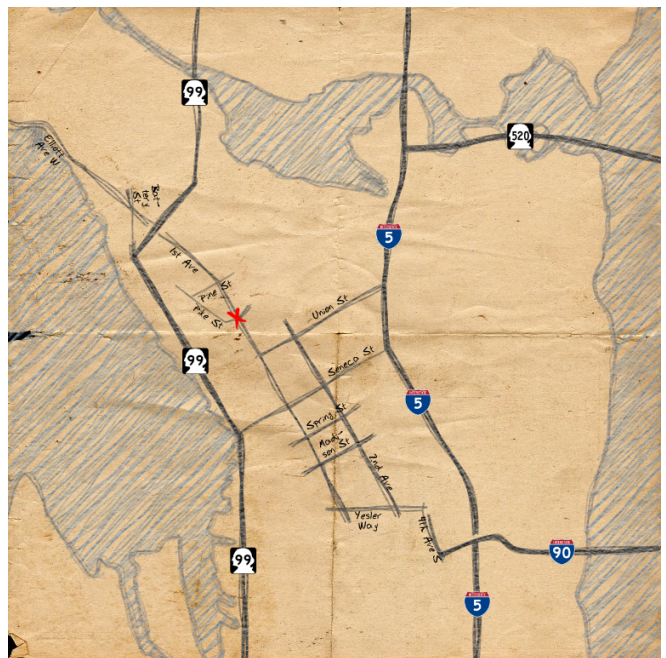
Princeton Result, Sketchy Style



St. George Result, European Style



Stanford Result, American Style



Seattle Result, Treasure Map Style

Figure 15: *Some results produced with our system.*



Figure 16: More results produced with our system.

- MACKANESS, W., AND BEARD, K. 1993. Use of graph theory to support map generalization. *Proc. CAGIS* 20, 4, 210–221.
- MERRICK, D., AND GUDMUNDSSON, J. 2006. Increasing the readability of graph drawings with centrality-based scaling. In *Proc. APVis*, Australian Computer Society, Inc., 67–76.
- PATEL, K., CHEN, M., SMITH, I., AND LANDAY, J. 2006. Personalizing routes. In *Proc. UIST*, 187–190.
- RAMER, U. 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing* 1, 2, 244–256.
- RAUBAL, M., AND WINTER, S. 2002. Enriching Wayfinding Instructions with Local Landmarks. *GIScience: Second International Conference* 2478, 243–259.
- SANDERS, P., AND SCHULTES, D. 2007. Engineering fast route planning algorithms. *Lecture Notes in Computer Science* 4525, 23.
- STEVENS, A., AND COUPE, P. 1978. Distortions in judged spatial relations. *Cognitive Psychology* 10, 3, 422–437.
- STOTT, J. M., AND RODGERS, P. 2004. Metro map layout using multicriteria optimization. In *Proc. Int. Conf. on Info. Vis.*, 355–362.
- THOMSON, R., AND RICHARDSON, D. 1995. A graph theory approach to road network generalisation. In *Proc. ICC* 17, 3–9.
- TVERSKY, B., AND LEE, P. 1999. Pictorial and verbal tools for conveying routes. In *COSIT*, C. Freska and D. M. Mark, Eds., 51–64.
- TVERSKY, B. 1981. Distortions in memory for maps. *Cognitive Psychology* 13, 3, 407–433.
- WINTER, S. 2002. Modeling costs of turns in route planning. *Geoinformatica* 6, 4, 345–361.
- WOLFF, A., 2009. The map-labeling bibliography. <http://illwww.itl.uni-karlsruhe.de/~awolff/map-labeling/bibliography/>.