Map Warping for the Annotation of Metro Maps

Joachim Böttger, Ulrik Brandes, Oliver Deussen, and Hendrik Ziezold ▪ *University of Konstanz*

When people use a city's public-transportation system, they face a seemingly simple task. They start at a point somewhere in the city, go to a nearby station, look for the best connection to another station near where they want to go, travel to that station, and finally go to their destination. Usually, people use two maps to accomplish that task: an ordinary geographic city map (a street-level map) and a schematic map of the public-transportation system (a metro map).

Schematic transportation maps usually contain little or no detail describing the environment of stations or their embedding in the surrounding area. The annotation of a distorted city map alleviates this deficiency and further improves the usability of schematic transportation maps by merging two different navigational spaces.

The reason for that is that both types of maps have advantages and disadvantages. The street-level map is well suited for gaining detailed information down to every single street. However, it struggles to give a fast overview of the public-transportation network, even if it includes that network. A metro map, on the other hand, is optimized for readability of connections and the transportation network's structure. However, it seldom shows the different stations in their surroundings, and it fails to deliver the needed contextual information for the task we just described. (For more on these two types of maps, see the "Navigation with Maps" sidebar.)

To ease the transition between metro maps and street-level maps, we developed a method that warps the street-level map information to fit a metro map. The result is a compound map containing both transportation network information and detailed street information. For that warp, we use a mapping from the field of image distortion, which is especially suited for geographical information. We also introduce features that yield dynamic interactive maps applicable for both street-level navigation and navigation in the public-transportation system.

Combined maps

One approach for combining metro maps with street-level information is *spider maps* (see sfcityscape.com/maps/spider.html and tfl.gov.uk/tfl/gettingaround/maps/buses), which are transfer guides for metropolitan areas. A spider map is a transportation map that centers on one station and displays the local area surrounding the station as a geographic street map, aiding the user when changing bus and metro lines. However, this approach requires adjusting the schematic layout to provide display space at the station of interest and its surrounding streets. Additionally, for someone starting at an arbitrary point in the city and having to choose a nearby station to walk to, a street-level map centered at this starting point is more useful than a street-level map centered at one station.

To alleviate these disadvantages, we propose annotating metro maps with all the information usually incorporated in street-level maps, without modifying their design. So, in a way, our approach is the opposite of annotated geographic maps.

Designers of metro maps sometimes include real-world items such as coastlines or rivers, but apart from that, these maps don't show most real-world features because their placement in the map isn't trivial. Owing to the metro map's lack of distance relations, the expedient positioning of street-level details requires an extrapolation of the deformations caused by the schematization. Today, many metro maps contain no detailed information other than the stations and their connections; they restrict themselves to describing the transportation system's navigational space as well as possible. Evidently, such maps are strongly specialized because

 Published by the IEEE Computer Society 0272-1716/08/$25.00 © 2008 IEEE

Navigation with Maps

In the main article, we concentrate on features of two types of maps that serve as navigational aids in metropolitan areas. Schematic maps of public-transportation systems (metro maps) are designed to effectively and efficiently convey possible itineraries. Street-level maps usually convey relative positions and distances of a wealth of locations. These very different purposes have led to differences in these maps' design.

Basically, maps represent relations between locations. Using this seemingly simple definition, we can easily distinguish the main characteristics between these two types of maps.

Street-level maps

One goal of detailed street maps is to minimize distortion, which means showing the real world in a way that's geometrically similar to what we would see from a vantage point high above a city. Accordingly, street intersections have the same angles they do in reality, and features such as parks and rivers have the same distinct shapes they have in the real world. This makes it easy to mentally put ourselves on the map and autonomously navigate through the city because we can use the shapes to find out where we are.

Although annotations are usually disproportionate (for example, real streets are often narrower than they appear on a map), distances and angles between geographic features aren't. The spatial relations in a street-level map are thus on an interval scale.

Street-level maps contain much more information than only the network of streets, such as landmarks, public facilities, and many other aspects of the surrounding environment. By representing geographical information in addition to this very high level of detail, street-level maps show a large variety of relations between locations. This abundance of detail is necessary because people use a city map for many different tasks. Most of these tasks require navigational decisions on a much smaller scale than the decisions people make when traveling in a public-transportation network.

Metro maps

Schematic transportation maps (metro maps) are designed to show the navigational information for the transportation system, preferably in a small space. In 1931, the pioneer of metro maps, Harry Beck, conceived his well-known map of the London Underground (the tube map)—a design landmark that forms the basis for metro maps today.[1] To achieve an expedient representation of the Underground, he had to make the central area appear larger, because the stations were closely crowded there. Drawing a map of the whole area in a limited space placed the stations in the center too close to each other for their connections to be distinguishable. Beck imagined he was using a convex lens to ensure readability in the center and in the periphery at the same time. He formulated the general design principle of placing all the stations at equal distances, although their geographical distances might be very different. This also reflects that metro travel time is approximately independent of the real-world distance between stations, because a train takes a relatively long time to drop off and pick up passengers and to accelerate and decelerate.

Another typical schematization requirement for metro maps is straightening the route lines by placing the stations of a line on straight lines, if possible. The overall shape is further simplified by restricting station positions to be at only a few discrete angles relative to each other, which, for example, leads to route lines being mapped to only verticals, horizontals, or diagonals. This makes it easy to mentally connect stations on the same line.

The resulting metro maps avoid intersections with small angles and are generally easily and intuitively readable. Automatic layout of metro maps has recently grown into an active field of research.[2] However, most metro maps are still manually fabricated by designers, who tweak the maps until they look just right and don't always strictly adhere to the aforementioned design principles. We use manually fabricated layouts as input for our method. One argument for using existing manually fabricated layouts is that a city's inhabitants are already familiar with them. We assume that, over time, people have adapted their mental map of the whole city to these layouts. Subsequently adapting to a different layout would impose mental strain. Nevertheless, our method also works with automatically generated metro maps.

To find a way from one station to another in a public-transportation system, the user requires only an overview of the connectivity between the stations. So, the type of relations on a metro map differs from that of a street-level map; easy perception of the existence of services and connections takes precedence over geographic accuracy. Metro maps preserve spatial relations on an ordinal scale, if at all.

References

1. K. Garland, *Mr. Beck's Underground Map*, Capital Transport Publishing, 1994.
2. A. Wolff, "Drawing Subway Maps: A Survey," *Informatik—Forschung & Entwicklung* [Computer Science—Research and Development], vol. 22, no. 1, 2007, pp. 23–44.

they can't reliably display geographically accurate information about the transportation system or its surroundings.

The essential feature of both street-level and metro maps is position indicators for station locations. Because schematic maps are designed to preserve stations' relative positions, it seems natural to use these positions for aligning a street-level map with a metro map.

Different approaches to explain how the two maps correspond exist. To analyze the distortions in map layouts due to schematization, Bernhard Jenny uses MapAnalyst, a tool originally developed for visualizing geographical errors in historical maps.[1] He annotates a metro map with visual hints to aid understanding of the implied distortion. Applying this to the London Underground map (the tube map), Jenny observes the typical features of metro map layouts. For example, in his scale isoline visualization, that map's fisheye character is clearly noticeable. Alexander Klippel and Lars Kulik introduce another approach for visualizing such distortion by applying it to the commonly known grid squares of a city plan.[2] Derek Reilly and Kori Inkpen use a slider to control the morphing between a street-level map and a tube map.[3,4] Coupled with alpha-blending, this map morphing makes correspondences obvious.

Because schematic maps are designed to preserve stations' relative positions, it seems natural to use these positions for aligning a street-level map with a metro map.

Facilitating an intuitive understanding of correspondences is a prerequisite in our approach, just like the avoidance of foldovers and occlusions (artifacts that occur if different parts of a warped map are mapped to the same place after a distortion). However, those prerequisites by themselves won't accomplish the integration of high-resolution, detailed street-level information and high-level, schematized network diagrams. This goal requires combining the appropriate level of detail with zooming techniques.

Method

In our warping method, the corresponding pairs of station positions serve as control points. The positions in the street-level map are the starting positions; the positions in the metro map are the end positions. This yields a mapping function that, when applied to the street-level map, shifts the stations to their positions in the metro map and distributes all the other real-world features smoothly between them. We then use this warped map to augment the metro map to support navigation and orientation in the parts of the real world between stations.

Detlef Ruprecht and his colleagues describe different methods for distorting 2D information.[5] All these methods solve the basic problem of warping: Given 2D information and a set of control points in this information, the goal is a mapping function moving these control points from their starting positions to arbitrary end positions. For satisfactory warping, the mapping function should be interpolating; that is, it precisely maps the control points' starting positions to their end positions. Furthermore, the mapping should be smooth; it shouldn't introduce discontinuities between the control points. Ideally, the mapping should also be free of overlap. In contrast to the distortion analysis and visualization by MapAnalyst,[1] avoiding overlap is important for warping geographic information to support navigation tasks. Otherwise, parts of the information completely disappear. The "Warping" sidebar (page 60) provides more detail on our method.

Prototype implementation

As proof of concept, we implemented a prototype system. For geographic information, we use US Census TIGER map data (Topologically Integrated Geographic Encoding and Referencing, tiger.census.gov), which contain vector data of detailed street information and landmarks such as water surfaces, parks, airports, and public institutions. These vector data can help provide good quality for rendered maps over a wide range of resolutions. They also provide the ability to transform topography independently from textual and symbolic labels to ensure readability. Moreover, these data are in the public domain and are sufficiently detailed to demonstrate our approach's potential for actual city plans.

We manually annotated the data with metro stations' geographic positions, compiling this information from other publicly available sources such as GoogleMaps (maps.google.com). Figure 1 shows a street-level map of the Washington, D.C., metropolitan area annotated with the metro stations' geographically correct positions, the corresponding metro map, and the metro map annotated with the warped street-level map.

To warp the street-level map, we first sample long lines in the original data sufficiently to avoid artifacts when rendering them and the polygons between them. This is necessary because although the mapping functions we described earlier are smooth, our approach maps straight lines to curves. Mapping only the start and end of a line and connecting these in the warped image again by a straight line generally don't produce smoothly deformed curves.

After subdivision, we evaluate the mapping function for every point of the geographic vector data, as we described in the section "Method." We neither use a grid with a fixed cell size nor rasterize our data beforehand, which is usually done when applying image-warping functions. Calculating the mappings is time intensive; our examples took approximately one hour to process on an ordinary desktop computer. However, we need to calculate and save the mapping only once for every set of control points.

To preserve quality, we render the distorted street-level data consisting of lines and polygons after the warp using OpenGL and GLUT (the OpenGL Utility Toolkit), reaching interactive frame rates. In the end, the metro stations, which were manually drawn into the street-level map, have the same positions as the stations in the metro map. We thus obtain a compound map showing topological and topographic information—the metro map annotated with the distorted street-level map.

A particularly nice feature is that our warping approach lets us interpolate mapping between exact geography and schematization. Placing stations in a convex combination of geographic positions and positions in the metro map yields a compromise between geography and schematization. We can extend this compromise to the entire compound map by linearly interpolating between the street-level map and its distortion based on the metro map. We sketch an important application of this feature in the next section.

Use cases, examples, and enhancements

We can envision three different use cases for our method. First, large, static depictions such as wall maps at metro stations could show more detail where needed, compared with an ordinary map.

Second, if display space is limited, a static overview over the compound map could provide annotation of the metro map focusing on large streets and landmarks but still aid rough orientation in the city. These two cases already make it obvious that we must address the level of detail.

The most interesting use case is the interactive application of our method for small displays, such as PDAs. Here, our method demonstrates the advantages of linking two navigational spaces, as we describe in the section "Interactive Warping Zoom."

We applied our method to maps of the Washington, D.C., and Boston areas. We chose these two cities because they contain typical features such as airports, lakes, rivers, coastlines, islands,

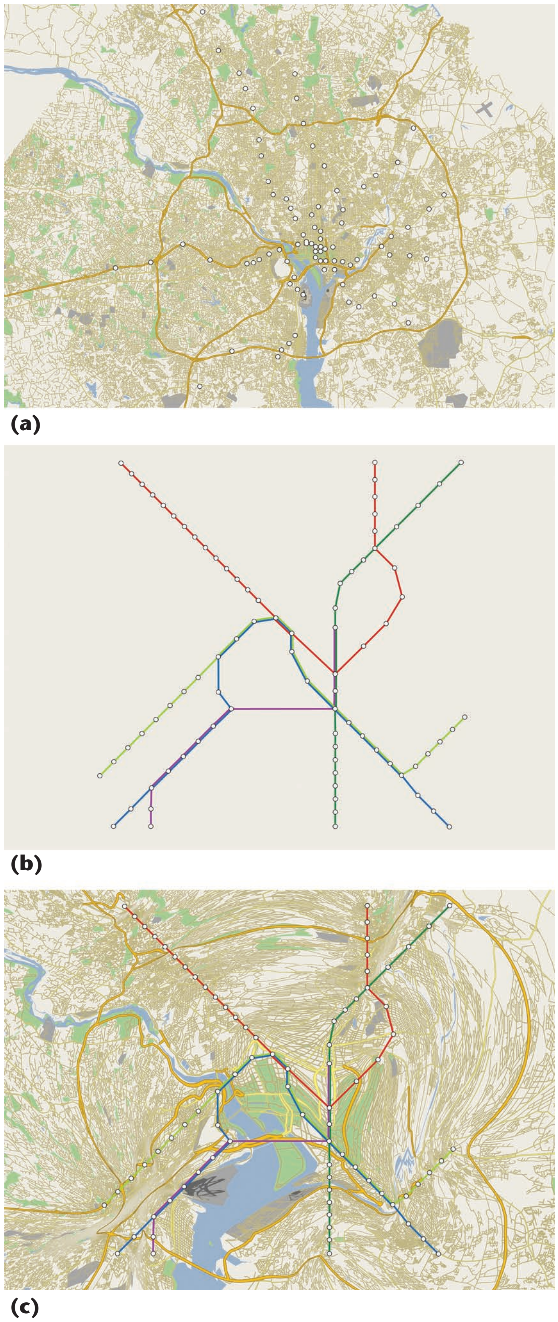

Figure 1. Maps of metropolitan Washington, D.C.: (a) a street-level map with the positions of metro network stations superimposed, (b) a metro map layout of the same area optimized for readability, and (c) our compound map, which annotates the metro map with a warped street-level map. (a) and (b) show our two input data sets; while it's possible to discern geographic details in the street-level map, these are completely missing in the manually fabricated metro map layout.

harbors, and parks and a fairly complex transportation system with nontrivial graph structure.

Figure 2 (page 62) shows street-level (left) and compound (right) maps of these cities' transporta-

Warping

For automatic integration of two corresponding maps, each one optimized for adequately displaying its respective navigational information, triangulation-based methods are inapplicable. This is because these methods suffer from foldover and other discontinuities, which aren't easily solved. (Foldover is an artifact that occurs if different parts of a warped map are mapped to the same place after a distortion.) So, we chose a warping method using scattered data interpolation that produces smooth, interpolating mapping functions. We also want to keep angles in the distorted map as similar to the real-world angles as possible, which keeps the shapes of real-world features recognizable.

Moving least squares

Scott Schaefer and his colleagues describe a moving-least-squares algorithm that interpolates a similarity transformation between the control points.[1] This way, angles are less distorted compared with interpolating only with general affine transformations.

Given a set of control points p, their position after the warping q, and an arbitrary single point v, Schaefer and his colleagues solve for the optimal affine transformation l_v that minimizes

$$\sum_i w_i \left| l_v(p_i) - q_i \right|^2$$

The method is called *moving-least-squares minimization* because the weights w_i depend on v:

$$w_i = \frac{1}{\left| p_i - v \right|^{2\alpha}}$$

The parameter α controls the decay profile for the distance and should be larger than 1. For our examples, we experimentally chose it to yield satisfying results; a typical value was 1.5.

This leads to a different transformation $l_v(x)$ for each v. Restricting the allowed transformations to similarity transformations, Schaefer and his colleagues find the following optimal mapping functions for each v:

$$l_v(x) = (x - p_*) \frac{1}{\mu_s} \sum_i w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} \left(\hat{q}_i^T - \hat{q}_i^{\perp T} \right) + q_*$$

Here, p_* and q_* denote the weighted centroids:

$$p_* = \frac{\Sigma_i w_i p_i}{\Sigma_i w_i}$$

$$q_* = \frac{\Sigma_i w_i q_i}{\Sigma_i w_i}$$

Furthermore,

$$\hat{p}_i = p_i - p_*$$

$$\hat{q}_i = q_i - q_*$$

$$\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T$$

and $\perp$ is an operator that maps a vector (x, y) to $(-y, x)$.

We apply these mapping functions for single points individually to the points in our geographical data sets. As Schaefer and his colleagues point out, the mappings still suffer from overlap. Figure A1 shows a simple example clarifying the resulting overall mapping functions by applying them to a regular grid. In Figure A2, the overlapping parts of the resulting 2D mapping function are clearly visible. Thus, to achieve overlap-free mapping functions, we combined this mapping method with the overlap control that Bernard Tiddeman and his colleagues described.[2]

Overlap control

Tiddeman and his colleagues observed that for any given mapping function, you can derive another mapping function by scaling the mapping—that is, interpolating it with the identical transformation. Such a scaling operation with a scaling factor s yields, for our case, this mapping function:

$$l_S(v, s) = (1 - s)v + s l_V(v)$$

Their other key observation is that overlap occurs at any point in a given mapping function if the determinant of its Jacobian changes signs. (The Jacobian is the matrix of the partial derivatives at each point.) So, this determinant J must be restricted to be at least positive. Because values of J close to 0 mean that the mapping at that point compresses the warped information strongly, Tiddeman and his colleagues restrict J further by requiring it to be larger than a minimal value $J_{\min}$.

To calculate J, we can use estimates of the partial derivatives by mapping two points close to a point v:

tion systems. For Boston (bottom), the compound map's center is greatly magnified compared to the surrounding area, similar to a fisheye lens. For Washington, D.C. (top), this effect is even more clearly visible in the compound map. The mapping functions manage to keep the areas close to the stations relatively undistorted but strongly stretch areas between the stations.

Level of detail

Because showing all the small details in a limited space can lead to indistinguishable visual clutter,

$$\left(\frac{\partial f}{\partial x},\frac{\partial g}{\partial x}\right)\approx\frac{I_v(v)-I_v(v+(\delta,0))}{\delta}$$

$$\left(\frac{\partial f}{\partial y},\frac{\partial g}{\partial y}\right)\approx\frac{I_v(v)-I_v(v+(0,\delta))}{\delta}$$

$$J=\frac{\partial f}{\partial x}\frac{\partial g}{\partial y}-\frac{\partial f}{\partial y}\frac{\partial g}{\partial x}$$

Here, δ is some small value. For several scaling factors s, with $0 < s < 1$, the resulting mapping function is guaranteed free of overlap.

To find an optimal scaling factor, we must solve the quadratic equation

$$J=\left(\left(s\frac{\partial f}{\partial x}+1\right)\left(s\frac{\partial g}{\partial y}\right)+1\right)-s^2\frac{\partial f}{\partial y}\frac{\partial g}{\partial x}=J_{\min}$$

for the Jacobian determinant to be $J_{\min}$. Solving a quadratic equation yields between two and zero roots. The Jacobian determinant is always equal to 1 at $s = 0$ and only gets smaller than the minimal value $J_{\min}$ at the equation's roots. So, the mapping is locally free of overlap or strong compression for all scaling factors larger than 0 but smaller or equal to the smallest root in the interval between 0 and 1. To ensure quick convergence, the method uses this root as the scaling factor. Because the control points shouldn't overshoot their destination, we use 1 if no such root exists.

To find an overall optimal scaling factor would require solving the equation for all points in the mapping function. Because this isn't possible, the equation is usually solved at discrete positions on a regular grid. We solve it for every point we map individually. Then, the overall best scaling factor is the minimum of all the locally optimal factors.

Scaling the whole mapping with the derived scaling factor yields a new mapping that doesn't fulfill the warp's constraints but already brings the control points closer to their destinations (see Figure A3). Iteration of the process and concatenation of the partial mappings brings the control points arbitrarily close to their destinations. However, this method doesn't guarantee convergence for all cases. Also, choosing a $J_{\min}$ that's too small leads to unnecessarily strong compression; choosing one that's too big prevents quick convergence. A typical value we used was 0.5. With this value, the overlap control worked well for several different examples, which converged within 5 to 15 iterations. Figure A4 shows the iterative process's result for our simple example.

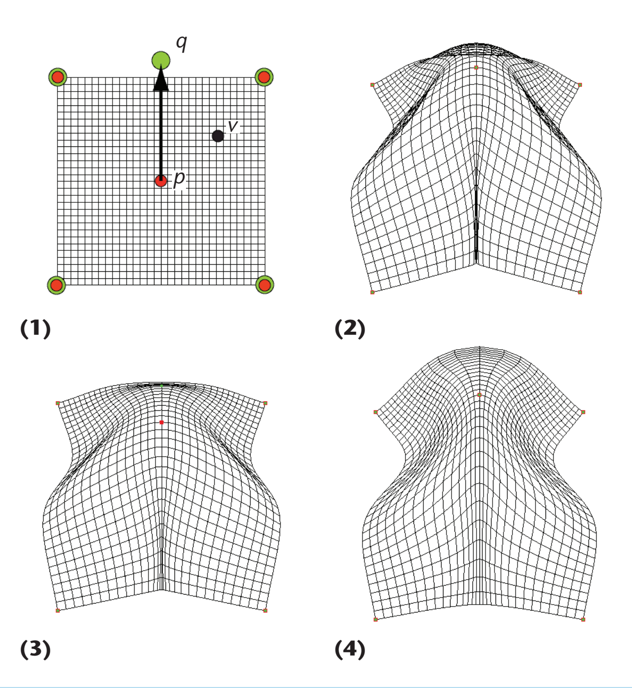

Figure A. (1) An undistorted grid with fixed control points at the corners and one control point moving the middle of grid *p* to a position outside grid *q*. (2) The middle-least-squares method results in overlap in the 2D mapping function. (3) Scaling the mapping yields a mapping function that moves the control point closer to its destination position. (4) Iterating this process and concatenating the partial mappings results in a mapping function fulfilling the constraints without overlap. The angles at the corner are still right angles after the mapping.

References

1. S. Schaefer, T. McPhail, and J. Warren, "Image Deformation Using Moving Least Squares," *ACM Trans. Graphics* (Proc. Siggraph), vol. 25, no. 3, 2006, pp. 533–540.
2. B. Tiddeman, N. Duffy, and G. Rabey, "A General Method for Overlap Control in Image Warping," *Computers and Graphics*, vol. 25, no. 1, 2001, pp. 59–66.

we must consider the local magnification and compression for depicting the geographic map's different features. During the iterative mapping, we estimate for each point the partial derivatives for overlap control. We can use the matrix of these partial derivatives, the Jacobian, for level-of-detail control because its determinant yields the local area magnification, and its condition number is proportional to the local compression.

We found it helpful to modify the thickness of linear features such as streets to be directly proportional to the local magnification and indirectly

Figure 2. Applying map warping. On the left are street-level maps of Washington, D.C. (top) and Boston (bottom) in geographically correct layout. On the right, those maps are fitted to their respective metro maps. Owing to the fisheye-like character of the mapping functions, we can now discern details in the cities' centers that aren't visible in the undistorted street-level maps.

proportional to the compression. This evenly distributes the density of features over the whole depiction.

Interactive Warping Zoom

Metro maps typically are too small for street-level navigation. To make the navigational information for the street level readable, we must either enlarge the annotated metro map to the size of a regular street-level map or enhance the compound map through interactive zooming.

Enlarging the compound map contradicts metro maps' main advantage, which is to give a quick overview of the transportation system, preferably on a small map. So, we implemented Warping Zoom. This interactive technique couples scaling of the map with a transition between our schematic compound map and the geographically correct compound map. To achieve this transition, we exploit the gradual-distortion technique we described earlier. While zooming, we interpolate between the distorted and undistorted map and simultaneously translate the map in a way that keeps the map's center at a constant position on the screen.

Figure 3 shows Warping Zoom's effect. The resulting dynamic compound map is especially applicable for mobile devices—because their display size is usually very small, dynamic maps with zooming functionalities are generally preferable. When zooming out of an interactive general city map, the user wants a quick overview of the city. When zooming in, the user wants detailed information about a specific region or point or wants to read navigational information for the street level.

People use public transportation systems to reach their approximate destinations. For such navigation, a metro map is adequate. So, when zooming out with Warping Zoom, the user gets a quick overview of the system—which is naturally a schematic layout.

Our implementation annotates this metro map with warped street-level information at an adequate level of detail. The transportation system stations are the interfaces between two navigational spaces: the transportation system and the street-level space.

Leaving the transportation system at a specific station, the user must navigate on the street level to reach the exact destination. So, the user requires geographically accurate information about the surroundings rather than a quick overview of the transportation system. In addition, the zoomed-in section would display just a few stations, thereby invalidating schematization's advantages. Thus, when the user zooms in on the compound map, it must be displayed in an undistorted and unschematized layout.

To make the interpolation between the distorted and the undistorted map feel intuitive, we define starting and ending scaling factors for the transition, taking into account the different maps' configurations and sizes. We display the undistorted compound map when only a few stations are visible. When the other end of the transition is reached, the whole schematized metro map fits in the display.

The level-of-detail control makes all steps of the transition readable. On the coarsest level, the metro map is annotated mainly with the most prominent features of the city, such as parks, rivers, and large roads. The smaller streets appear more clearly when the user zooms in and wants to navigate the street network.

Adapted fisheye views

Although Warping Zoom enables the examination of small geographic details, it's also desirable to show magnified geographic detail and schematized network information in one seamless view, as in spider maps. However, mixing the two spaces by simple spatial interpolation potentially leads to overlap because overlap's absence is guaranteed only for the concatenation of overlap-free mappings. So, to guarantee seamless transition between detail and context, we chose a different approach: we apply fisheye mappings to the schematic space.

Using an ordinary fisheye would make only local compression inherent in the warped information more visible. To counter this compression, we calculate the compression factor using estimations of the partial derivatives at a center of interest. These define a locally linear transformation, and using a singular value decomposition, we can estimate the magnification factors in the different directions. If these factors aren't equal, the information is compressed at that point. To counter that compression, before applying an ordinary fisheye mapping as T. Alan Keahey and Edward Robertson do,[6] we apply a mapping that stretches the area around the center of interest. We then let the stretching decay in proportion to that center's distance. Beyond a certain radius from the center, our mappings leave the schematically distorted information unchanged.

The resulting mappings are free of overlap and stretch the warped information just enough to locally guarantee angular faithfulness. Figure 4 (next page) illustrates our method. Figure 4a shows the geographically correct shape of an airport in Washington, D.C. Magnifying the respective detail in the schematized view using an ordinary fisheye view (see Figure 4b) exhibits a compressed shape with a changed aspect ratio. Figure 4c shows our technique's results: the airport is larger and close to its original shape. The red circle in Figure 4b corresponds to the ellipse in Figure 4c, illustrating the difference between the ordinary fisheye and our method.

Figure 5 (next page) shows the whole compound map of Washington, D.C., in schematic layout. Our fisheye technique magnifies the area around a point of interest. Such an interactive map can be used like spider maps for combined navigational tasks.

Distance information

To help understand geographical-distance relations in the schematic view, we annotate the metro map with isolines at certain distances from the closest stations. During the iterative mapping, we distort the points of a regular grid in addition to the street-level data. To this end, we apply the mapping

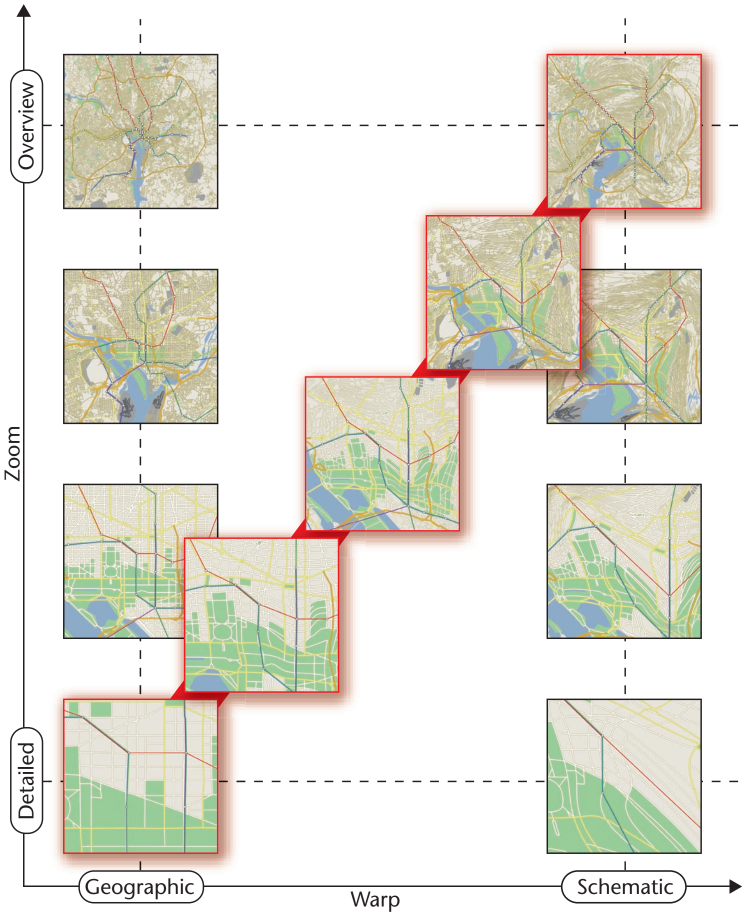

Figure 3. Warping Zoom (here shown on the diagonal in red) combines zooming and warping between the schematized and geographically correct map. This makes it possible to employ the schematized layout for an overview and a detailed geographical layout for localization and street-level navigation. Warping Zoom lets users discern the connections in the center of Washington, D.C, which aren't discernible in the geographic overview (top left) at this resolution.

function to the single grid points, which results in a distorted grid as in Figure 6 (page 65).

We can then visualize real-world distances from points on the map to the next station. First, we calculate the distance of every regular grid point to the closest station. Then, we apply a marching-squares algorithm to the distorted grid to yield isolines denoting equal distances to the closest station.

For the undistorted grid, these isolines consist of circular shapes centered around each station. After the distortion, the shapes are more complex, as Figure 7 (page 65) shows. For example, two stations, which are close to each other, are connected by an hourglass-like structure. Moreover, a gap between these structures indicates large distances between the corresponding stations.

This way, for example, our compound maps can help users find the station nearest to a specific destination.

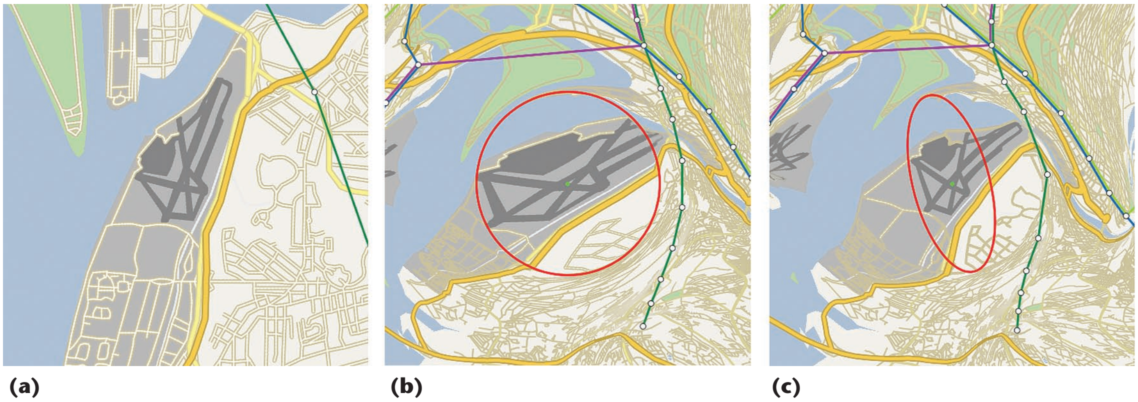

Figure 4. Mapping a Washington, D.C. airport: (a) the geographically correct shape of the airport, (b) an ordinary fisheye view applied to the metro map, and (c) an adapted fisheye view. The airport appears rotated compared with the geographic layout but retains its distinct shape.

Our method has some shortcomings. First, it's not clear whether the mapping functions are optimally suited for human perception of geographical information. They stress the importance of keeping angles locally intact, maybe at the expense of readability for extreme distortions.

Although we think existing handmade metro map layouts are superior to automatically generated ones, merging our method with one for automatically generating metro map layouts seems promising. This way, we might find an even better compromise between detail and schematization, avoiding extreme distortions in the street-level map and hard-to-read configurations in the superimposed schematic representation.

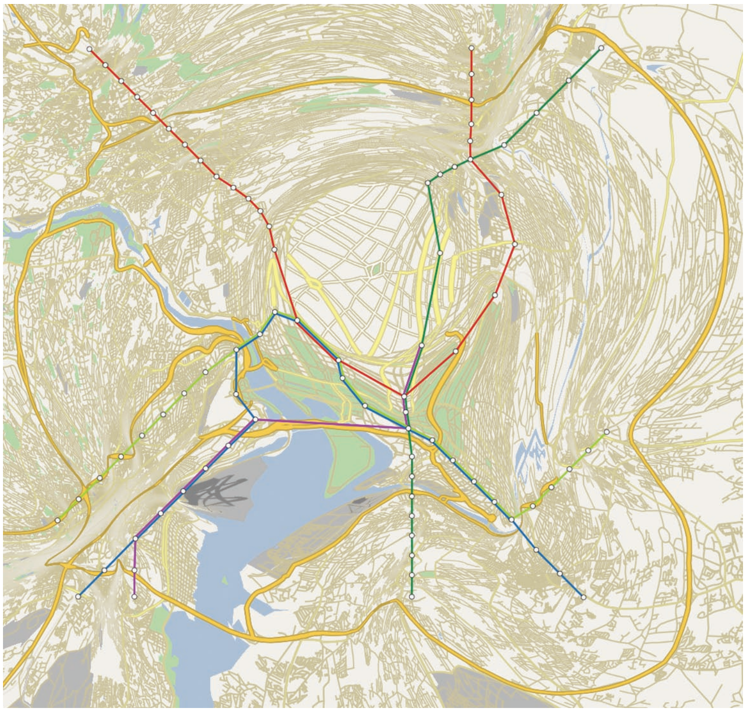

Figure 5. A compound map of Washington, D.C. in a schematic layout, with an area around a point of interest magnified and stretched by our fisheye technique. With the local compression being countered, the intersections close to the center of interest keep their right angles.

We know our renderings don't possess all the features of commercial city maps. We haven't addressed or implemented more advanced algorithms for rendering labels, symbols, and additional data sources. However, their application is independent from our technique. Our purpose here isn't to discuss a fully featured map generator but to demonstrate the utility of our method for combining schematic and geographic maps.

We also believe there are other opportunities to apply our technique in graph visualization, annotation of route sketches, and similar fields.

Acknowledgments

The Volkswagen Foundation partially supported this research under grant II/81425 (the visual-navigation project), and the Deutsche Forschungsgemeinschaft supported it under grant GK-1042 (the Explorative Analysis and Visualization of Large Information Spaces research training group). We thank Steffen Bogen, Gesa Henselmans, Albert Kümmel-Schnur, Felix Thürleman, Boris Neubert, and Johannes Kopf for fruitful discussions and helpful comments.

References

1. B. Jenny, "Geometric Distortion of Schematic Network Maps," *Bull. Soc. of Cartographers*, vol. 40, 2006, pp. 15–18.
2. A. Klippel and L. Kulik, "Using Grids in Maps," *Proc. 1st Int'l Conf. Theory and Application of Diagrams* (Diagrams 00), Springer, 2000, pp. 486–489.
3. D.F. Reilly and K.M. Inkpen, "Map Morphing: Making Sense of Incoherent Maps," *Proc. Graphics Interface*

2004, Canadian Human-Computer Communications Soc., 2004, pp. 231–238.
4. D.F. Reilly and K.M. Inkpen, “White Rooms and Morphing Don’t Mix: Setting and the Evaluation of Visualization Techniques,” *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 07), ACM Press, 2007, pp. 111–120.
5. D. Ruprecht, R. Nagel, and H. Müller, “Spatial Free-Form Deformation with Scattered Data Interpolation Methods,” *Computers and Graphics*, vol. 19, no. 1, 1995, pp. 63–71.
6. T.A. Keahey and E.L. Robertson, “Techniques for Non-Linear Magnification Transformations,” *Proc. 1996 IEEE Symp. Information Visualization*, IEEE Press, 1996, pp. 38–45.

Joachim Böttger *is a PhD candidate in the University of Konstanz’s Department of Computer and Information Science. He’s a member of the Deutsche Forschungsgemeinschaft Explorative Analysis and Visualization of Large Information Spaces research training group. His research interests include computer graphics, cartographic mappings, and detail-in-context visualization. Böttger received his diploma in computational visualistics from the University of Magdeburg. Contact him at joachim.boettger@uni-konstanz.de.*

Ulrik Brandes *is a professor of computer science at the University of Konstanz. His research interests include algorithmic graph theory, graph drawing, social-network analysis, information visualization, and algorithm engineering. Brandes received his habilitation from the University of Konstanz. Contact him at ulrik.brandes@uni-konstanz.de.*

Oliver Deussen *is a professor of computer science at the University of Konstanz. His research interests include analog and computer-generated graphics and information visualization. Deussen received his PhD from the Karlsruhe University of Technology. Contact him at oliver.deussen@uni-konstanz.de.*

Hendrik Ziezold *is a PhD candidate in the University of Konstanz’s Department of Computer and Information Science and a member of a visual-navigation project funded by the Volkswagen Foundation. His research interests include navigation and cartography, operation-centered design, and computer graphics. Ziezold received his diploma in computational visualistics from the University of Koblenz-Landau. Contact him at hendrik.ziezold@uni-konstanz.de.*

For further information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.

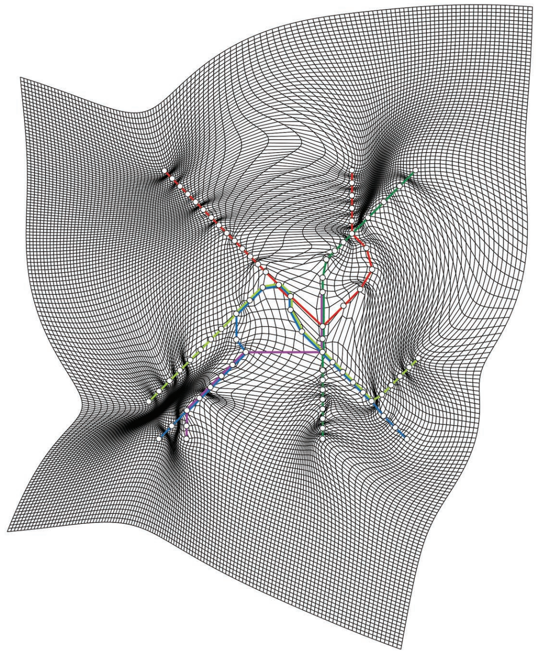

Figure 6. A regular grid distorted with the same mapping as the one used for warping the Washington data in Figure 2. We use the grid to render isolines around the stations to aid understanding of the geographical-distance relations.

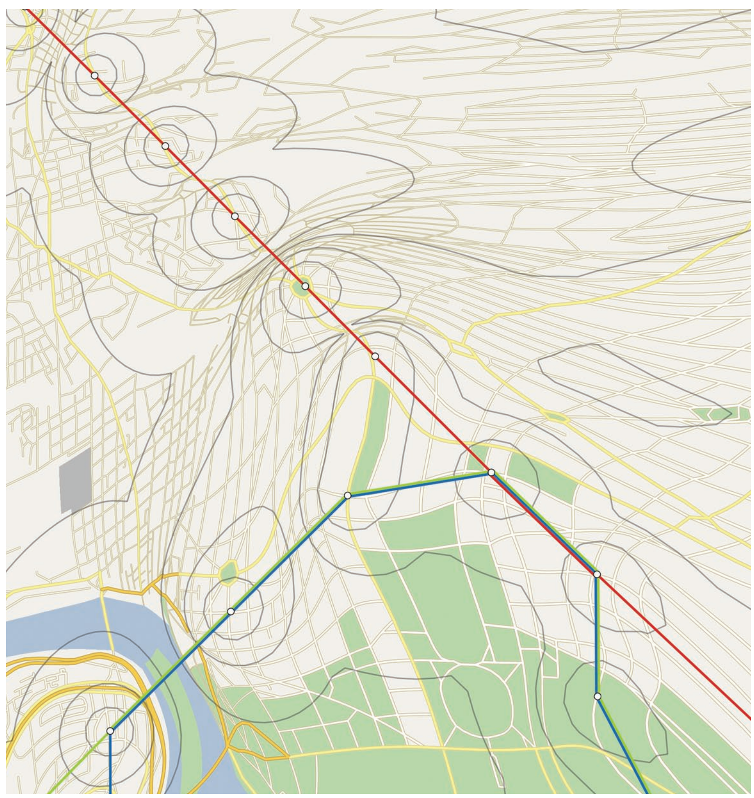

Figure 7. Isolines around stations. The lines are at a constant geographic distance to the closest station. So, nearby stations are connected by blob-like shapes, while gaps between these shapes indicate large distances. These structures make it possible to estimate the real-world distances in the distorted map.