International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:6, 2010

Automatic map simplification for visualization on mobile devices

Hang Yu

Abstract—The visualization of geographic information on mobile devices has become popular as the widespread use of mobile Internet. The mobility of these devices brings about much convenience to people's life. By the add-on location-based services of the devices, people can have an access to timely information relevant to their tasks. However, visual analysis of geographic data on mobile devices presents several challenges due to the small display and restricted computing resources. These limitations on the screen size and resources may impair the usability aspects of the visualization applications. In this paper, a variable-scale visualization method is proposed to handle the challenge of small mobile display. By merging multiple scales of information into a single image, the viewer is able to focus on the interesting region, while having a good grasp of the surrounding context. This is essentially visualizing the map through a fisheye lens. However, the fisheye lens induces undesirable geometric distortion in the peripheral, which renders the information meaningless. The proposed solution is to apply map generalization that removes excessive information around the peripheral and an automatic smoothing process to correct the distortion while keeping the local topology consistent. The proposed method is applied on both artificial and real geographical data for evaluation.

Keywords—Map simplification, visualization, mobile devices

I. Introduction

THE explosive growth of mobile Internet devices such as PDAs or smart phones brings about revolutionary changes for the lifestyle of people. These devices provide much convenience to people wherever they are and whenever they want. From the Internet they can obtain timely information on events relevant to their locations, such as finding the restaurants nearby or the nearest parking spaces when they are driving. Although these mobile devices quickly gained access to the Internet applications, there are some typical characteristics of the devices that they usually have restricted computing resources and small displays. These characteristics may reduce the likelihood of these devices being popularly applied. Let us take an example of car navigation system as one of the location-based services. In this scenario, the driver may only have very short time on checking the route map before making any decision. If a large area is entirely presented on the small display, the driver may only see an overview of his/her location without sufficient details. While a zoom-in function can give a remedy, the driver may lost the context information regarding to the environment. This leads to a challenge: how to render large data-set in a relatively small display. This is known as the "maximize the usage of screen real estate problem" in visualization.

Hang Yu is with the Institute for Infocomm Research (I^2R), A*STAR, Singapore, 138632, e-mail: hyu@i2r.a-star.edu.sg.

To deal with the issue of small displays, an effective solution is to represent geographical information in multiple scales, with finer scale at the point of interest. When merging variable scaled geographical information into a single image, the viewer is able to focus on the interesting region, while having a good grasp of the surrounding context. This is essentially visualizing a map through a fisheye lens. Usually the regions near the location of viewers are shown in a larger scale and those far away are shown in a decreased scale. By this means, the information from different scopes is fitted together as a single event for viewers.

However there is a disadvantage for the method that the map may be filled with excessive information on a small display which may disturb viewer to grasp the most relevant knowledge. For example, in driving navigation, the sinuosities along the route, although reflecting the actual roads condition, may cause information clutter to the viewer.

As for the problem on information cluttering, simplification techniques, including distortion and abstraction, are designed to improve the readability of the map [10], [11]. For most work in this field, the main objective is to remove less important details while salient features are retained. One drawback of such simplification is that it usually lacks aesthetic consideration as sharp turns may appear. Thus it is expected that the map is represented by a set of succinct and smooth curves with topological fidelity.

The paper is organized as follows. Section II reviews related work about analysis and visualization on mobile devices. Section III gives the proposed method which consists of three steps. Step 3 is the focus of the method, thus the detailed explanation is given in section IV. The implementation and experimental results are given in section V. Section VI concludes the paper.

II. Related work

Much research work has been devoted in the field on employing geographic information on small mobile displays. Generally, these work can be divided into three categories: visualization by interactive queries; visualization by map generalization and visualization by variable-scale representation.

Some researchers have proposed approaches for exploratory analysis of geographic data on mobile devices. These approaches is inspired by a method called dynamic queries [2], [13] which are typically applied in desktop PCs. The main idea is that by specifying desired values, viewers can have a fast and easy way to learn the interesting properties of the data. An initial query can be incrementally revised by

interactively imposing constraints on the results. However this approach has some disadvantages when applying on mobile devices with small displays [14]. Without showing objects whose attributes are not within viewers' constraints, viewers may lose context information regarding to the queries. To counter this problem, Burigat et al. [5] gave a visualization technique to help viewers to obtain contextual information. In their work, icons representing the degree that the results satisfy viewers' queries are placed on a map. By this way, viewers may have a more comprehensive view of the explored dataset.

Generalization is a common technique used in geovisualization to create appropriate views of a map for different scales. This technique has been applied in spatial data visualization on small mobile devices [4], [7]. The purpose is to remove excessive information or simplify objects so that viewers will have an easy access to relevant information to support their tasks. However, a generalized map at a fixed scale still may not meet the requirements for small device display as the generalized geographic objects may be too large to display such as a long and straight road. To solve this problem, shape simplification and distortion methods are employed. In the work by Agrawala et al. [1], geographic objects are generalized differently based on cognitive psychology research. For e.g., all the turning points on a route are more important than the exact length, angle and shape of the route. Similar techniques can be found in the work by Dong et al. [6]

Besides these achievements, there is research work specifically focused on the representation of geographic data on mobile displays. To adapt the content of maps onto the small mobile devices, Harrie et al. [8] presented a variable-scale method based on the principles of the Fisheye view [12]. The contents close to viewers' location are displayed in a larger scale with greater detail while others are in less detail. For the small scale part of the map, some operation is needed to increase the readability as there might be geometrical distortion due to the uneven distribution of the data objects. A map generalization method including building simplification and selection was employed that selected spatial objects were simplified while others were rejected. In the work by Li [9], the contents of a map are divided into different groups which are called regions of interests (ROIs). Geographic objects are selected from these ROIs by certain rules to present at different levels of details to viewers.

III. Proposed method

In this paper, instead of working on various objects representation, polygonal line or polyline is explored, which represents road or coastline. Other objects like labels, county boundaries, buildings, etc are not considered.

The following steps are taken to generalize a map and the map is rendered in a small display.

- Step 1: Objects outside of the ROI that have no relationship with objects in the ROI are discarded.
- Step 2: A fisheye lens transformation is applied on the retained objects from previous step.
- Step 3: The fisheye transformed map is smoothed.

The focus of the proposed work is on step 3. Given a collection of polylines $\mathcal{L}$, the intention is to find smoothed lines that achieve optimality while satisfying topographic constraints. The optimality lies in two aspects: to smooth the lines, and to reduce the derivation from the original. Formally, the optimization process is measured by a score function subjected to certain constraint as follows:

$$min : E = \sum_{l \subset \mathcal{L}} (E_s(l) + E_d(l)) \quad (1)$$

where $E_s(l)$ is the sum of curvature along a polyline l and $E_d(l)$ is the derivation of smoothed l from its original.

The curvature of a smooth curve can be defined as the inverse of the radius of the inscribed circle at each point along the curve. For the definition of curvature on polyline by the connection of discrete data points, one possible way to estimate the curvature of a point is to compute the inverse of the radius of the circle passing through the point and its two neighbours. As shown in Figure 1 (a), the curvature of a point P_i in the polyline can be given as $k(P_i) = 1/r$ where r is the radius of the circle that passes through 3 consecutive points P_{i-1}, P_i and P_{i+1}. In the implementation, the length of the polyline is simply used to represent the sum of the curvature along the polyline as intuitively the length can reflect the bending extent of the polyline.

The other item $E_d(l)$ can be measured using Hausdorff distance. However, due to the large number of points involved in the computation, $E_d(l)$ is approximated by the area exchanged between two polylines. For example, in Figure 1 (b), the derivation between the polyline and the smooth curve represented by dotted line is measured by the area difference under these two.

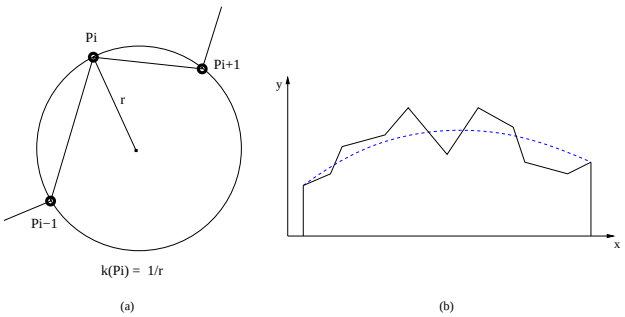


Fig. 1. (a): Curvature of a point in a polyline. (b):Derivation between a polyline and the smooth curve.

The topological constraint is:

- $\mathcal{C}$: the areas enclosed by polylines are preserved.

Note that $\mathcal{C}$ implies that no intersection or self-intersection among the polylines are created or removed.

In the following, a brief description of step 1 and step 2 are given. In Section IV, a detailed description of step 3 is given.

A. Objects filtering and fisheye transformation (Step 1 and 2)

1) Filtering non-related objects: In this step, the map objects having relationship with ROI are retained and the rest are discarded. In the implementation, the ROI is denoted as

a circular region. If an object has overlapped with the ROI, it is considered to be relevant to current navigation task. For example, in Figure 8 (a) an example is given showing a route map marked with a ROI.

2) Fisheye view mapping function: There are two main variations of fisheye view mapping: Cartesian and Polar transformation. Cartesian transformation is applied in rectangular coordinates while Polar transformation is applied in polar coordinates. The difference is that the former transforms data independently on X and Y directions while the latter transforms data in the radial direction originating from the focus.

Normalized polar fisheye transformation is employed as the distortion technique. Under this variation, a point $p(x, y)$ in rectangular coordinates is represented by its normal coordinates (r_{norm}, θ) with the focus $p_f(x_f, y_f)$ as the origin where $r_{norm} = ||p - p_f||$ and $\theta = atan(\frac{y-y_f}{x-x_f})$. The relationship between a point's polar coordinates (r_{norm}, θ) and its fisheye coordinates (r_{feye}, θ) is given according to the following distortion function:

$$r_{feye} = r_{max} * G(z, d); \tag{2}$$

where $G(z, d) = \frac{(d+1)*z}{d*z+1}$ and $z = \frac{r_{norm}}{r_{max}}$. d is the distortion factor that controls the intensity of the distortion and can be controlled through input from viewer. r_{max} is the maximum bounding value of radius r_{norm} along the direction of θ. θ remains constant during the transformation.

IV. Line smoothing (Step 3)

A. Main idea

Finding the global optimal solution of E.q. 1 seems to be difficult. Instead, a heuristic is given that iteratively local smoothing is performed. Each "local smoothing" essentially finds two Bézier curves that satisfy the constraint $\mathcal{C}$.

The set of polylines is smoothed one by one. For each polyline, it is randomly divided into sub-polylines. The sub-polylines are divided in a way that the smoothing problem on the set of polylines can be reduced to a sequence of simplified instances. To ensure that there is a smooth transition from a sub-polyline to the next one, during the smoothing of a sub-polyline, additional constraints are included on the tangent of the starting and ending points of the sub-polylines.

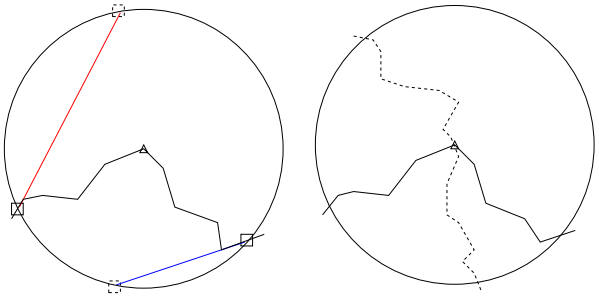
A complication appears to process open curves. Note that there is no "enclosed area" under an open curve, and it is not clear on how to impose the area preserving constraint on it. This will be discussed in Section IV-D.

B. Algorithm flow

The overall flow of the algorithm is as follow:

1) Pick a non-smoothed polyline l from the collection $\mathcal{L}$.
   a) Randomly pick a point p_c from l. Noted that the p_c is either the node of the l or the intersection point resided in the l. Find a circle with p_c as the center that the set of sub-polylines S from $\mathcal{L}$ covered by the boundary of the circle should all pass through p_c. The radius of the circle is obtained to the largest extent. The relationship between S and the circle belongs to the following two cases:
   - Case 1: The center of the circle p_c is the node point of the polyline. There is only one sub-polyline in S.
   - Case 2: The center of the circle p_c is the intersection point of polylines. There are more than one sub-polylines in S that pass through the center p_c. In the current implementation, only two sub-polylines are considered. It can be easily extended to more sub-polylines passing through the intersection point.

   These two cases are illustrated in Figure 2. Figure 3 gives a snapshot of the moving circle along two polylines. The radius of the black moving circle is obtained at the largest extent.



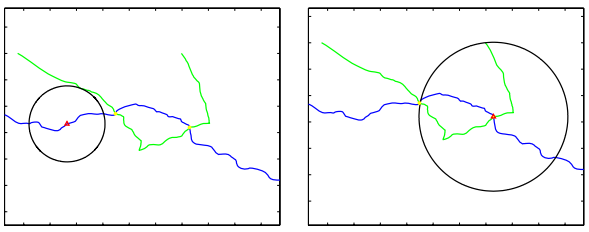
(a) Case 1 (b) Case 2

Fig. 2. The relationship between the line segments and the circle. The center of the circle is marked by a triangle symbol.

   b) For each sub-polyline $s_i \subset S$, perform a sub-problem to find the smoothed curve.
2) Process next polyline which is not smoothed.

C. Local smoothing in the sub-problem

The sub-problem is to find the smoothed curve(s) which minimize E.q. 1 under the constraint $\mathcal{C}$ for the two cases given above. Area-preserving Bézier curve fitting is employed for the smoothing process. Generally, a Bézier curve can be



(a) (b)

Fig. 3. The snapshot of the moving circle along two polylines. The yellow dots represent the intersection points of the two polylines. The red triangle represents the center of the moving circle.

determined by a number of control points. It is very convenient to adjust the shape of the curve by moving these control points. The advantage is that it is easy to regulate the area under the curve.

Figure 2 (a) shows the smoothing process for case 1. The process consists of the following three steps.

1) **Locating the control points.**
Four control points $P_1(A_x, A_y)$, $P_2(B_x, B_y)$, $P_3(C_x, C_y)$, $P_4(D_x, D_y)$ are used to determine a Bézier curve. Parametrically, the Bézier curve is represented as follow:

$$\begin{aligned} x(u) = \quad & A_x(1-u)^3 + 3B_x(1-u)^2u + \\ & 3C_x(1-u)u^2 + D_xu^3 \\ y(u) = \quad & A_y(1-u)^3 + 3B_y(1-u)^2u + \\ & 3C_y(1-u)u^2 + D_yu^3 \\ u \subset [0,1] & \end{aligned} \tag{3}$$

The coefficients in the equation represent the coordinates of the control points. Points P_1 and P_4 are the two end points of the Bézier curve and are obtained as the intersection points between the polyline and the circle. Points P_2 and P_3 are the two intermediate points. The adjustment of these two points will affect the overall shape of the obtained Bézier curve. These two points are obtained by extending the line segment from the polyline that intercepts the circle until getting another intersection with the circle. These extended line segments are marked as red and blue in Figure 2 (a). The intersection points are represented as a square with solid line and dashed line separately. Noted that the dashed-line square is obtained from the extension. For the solid-line squares, they are fixed as control points and decide the tangent vector at the two ends. The other two control points are chosen from the red and blue lines by some sampling interval.

2) **Obtaining Bézier curve under constraints.**
The obtained Bézier curve through fitting on the control points located in the first step should not cross over the circle and the new partition made on the area of the circle should be the same as that divided by the original line segments. The first requirement has insured that no imported intersection points occur and the second one is for the area preserving constraint. In the implementation, it is considered to be acceptable if a enclosed area is not altered substantially. Thus a threshold γ is set as the limitation for the percentage of area difference between a smoothed polyline and its original.

3) **Finding the optimal solution.**
There may be more than one Bézier curve obtained in step 2. These curves are the candidates for choosing the optimal solution judged by E.q. 1. Considering the relatively small parameter space, exhaustive search is applied to find the optimal Bézier curve.

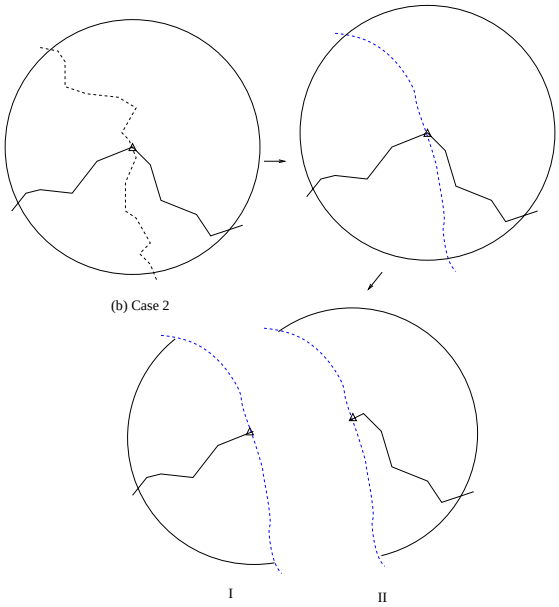


Fig. 4. The smoothing on Case 2. It is referred to as Case (b) in Figure 2.

For smoothing on case 2 that there are two sub-polylines in the safe region, following algorithm is given.

There are two steps for the algorithm. First step is to process one sub-polyline with the whole circle as the boundary. The problem can be reduced to case 1. As shown in Figure 4, the blue dotted line represents the Bézier curve fitting for the dotted polyline in Case 2. The following step is to process the two parts of the other sub-polyline. The smoothing of each part can also be reduced to case 1. To handle multiple polylines, the center needs to be fixed after the first step.

D. Area-preserving on open curves

To handle the area-preserving constraint on open curves, the idea of "diff-area model" in the work by Bose et al. [3] is adopted. This model gives a solution to simplify a polygonal path with the objective that the area above (or below) the path changed by the approximation is preserved. However their work is constrained to handle x-monotone polygonal path. Here a simple overview of the model is given.

Problem definition. Suppose P and Q are two x-monotone polygonal paths where Q is the approximating one of P. Let $\Omega_A(Q)$ be the area above P and below Q, and $\Omega_B(Q)$ be the vice versa.

Diff-area model. To measure the quality of the approximation by Q, diff-area model suggests that the cost function is $|\Omega_A(Q) - \Omega_B(Q)|$ which is the area exchange above (or below) P and Q. This model implies that the area is "exchangeable", that is, area loss at one location can be redeemed from another location.

In the algorithm, each polyline has been divided into a set of sub-polylines such that the smoothing problem is easy to handle for each sub-polyline. This division has also created the flexibility to apply diff-area model in each sub-polyline.

As each sub-polyline is smoothed in a circle locally found, applying diff-area model is identical to preserve area of the circle partitioned by the sub-polyline. The overall area-preserving constraint is satisfied while the local smoothing process is performed iteratively.

V. Implementation and experiments

The proposed algorithm is implemented on a synthetic data-set and a real map data-set.

Figure 5 (a) gives the full view of the synthetic data-set with 9 polylines in ROI are highlighted. The ROI is simply denoted by a black circle. All the polylines outside the ROI or the black circle are regarded as non-related to current navigation interest and filtered. The retained 9 polylines are depicted by different colors in Figure 5 (b). For the purpose of easy illustration, each route is marked by a sequence number at one end of the route.

Figure 6 gives the experimental results of the algorithm with different parameters. Figure 6 (a) and Figure 6 (b) give the fisheye transformation with different focus of interest. The distortion factor d is 3 and the focus is marked as a red dot. Figure 6 (c) and Figure 6 (d) give the smoothing results on Figure 6 (a) and Figure 6 (b). The threshold γ for measuring the area difference is set as 0.01. Figure 6 (e) and Figure 6 (f) have the same parameters as Figure 6 (c) and Figure 6 (d) except that γ is set as 0.3.

Note that the routes after smoothing have much less sinuosities comparing to their original counterparts. Meanwhile, the area enclosed by the routes are not changed drastically. Although the proposed algorithm performs smoothing locally, the global smoothness for the routes are achieved. Taking polyline 8 in Figure 6 (e) as an example, the 8 segments partitioned by the 7 intersection points not only stay smooth separately but also keep continuous on the conjunction points. Increasing γ may achieves better smoothing effect while the enclosed area is altered significantly.

Figure 7 gives the overall score function E for polyline 1 in Figure 6 (c). As the area difference is considered to be negligible, this curve reflects the curvature variation at each iteration. It shows that the solution provided by the proposed algorithm always improves the optimality of the route.

The real map data-set is extracted from a simple representation of the major roads in the state of Connecticut, US. It depicts the highway network in the state at 1:250,000 scale. Figure 8 gives the results on the algorithm.

VI. Conclusion

In this work, a map generalization algorithm is designed to handle visualization of vector-based map on devices with small display window. The algorithm firstly filters out non-related information for current navigation task. The following steps are applying fisheye lens to exaggerate objects in the ROI and smoothing the lines to remove clutter caused by the distortion. The main focus is on the line smoothing part. An algorithm is presented to formulate the smoothing process as an optimization problem which minimizes the overall curvature while preserving the enclosed area by the lines. A heuristic method is given to find the optimal solution by dividing the problem into the combination of a set of sub-problems. For each sub-problem, the objective is to find at most two Bézier curves given at most two polygonal lines which intersect at most once in a circular region, meanwhile the partitioned area of the region by the two lines are preserved. Experimental study demonstrates that the proposed algorithm achieves the approximation of global optimality for the generalized map.

There are three aspects to extend current work.

1) In each sub-problem, the Bézier fitting for each polygonal segment is determined by a small number of parameters. Thus the whole polyline could be represented by a compact format. This could be used in remote visualization by transmitting only the compact format of a map thus improving the efficiency. However, as the sub-problems are independent for each other, consideration should be given on how to reconstruct the correct final presentation upon receiving the transmitted parameters.
2) Currently, consideration is only given on the geometric features of a vector-based map. Generalization may become complex when the other features are considered, for example, the labeling of map objects. The positioning of the label will be a challenging issue when the corresponding object is relocated.
3) The efficiency of the algorithm can be improved if the process of choosing segments to smooth is controlled. There are a few crucial points on each polyline that dominant the major bending energy. The detection of these points is critical to accelerate the smoothing iteration.

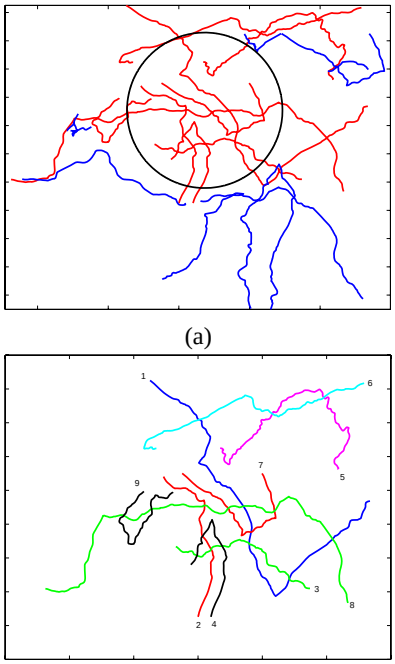


Fig. 5. The synthetic data with 9 polylines in ROI. The 9 polylines are depicted as red color while the rest are in blue. The ROI is denoted as the black circle.

ACKNOWLEDGMENT

The author would like to thank Editor Dr. B. Brojack and the anonymous reviewers, whose invaluable help for the publication of the paper is much appreciated.

REFERENCES

[1] Maneesh Agrawala and Chris Stolte. Rendering effective route maps: improving usability through generalization. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249, New York, NY, USA, 2001. ACM.

[2] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM.

[3] P. Bose, O. Cheong, S. Cabello, J. Gudmundsson, M. V. Kreveld, and B. Speckmann. Area-preserving approximations of polygonal paths. *Journal of Discrete Algorithms*, 4:554–566, 2006.

[4] Claus Brenner, Claus Brenner, and Monika Sester. Continuous generalization for small mobile displays. In *In International Conference on Next Generation Geospatial Information*, 2003.

[5] Stefano Burigat and Luca Chittaro. Visualizing the results of interactive queries for geographic data on mobile devices. In *GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 277–284, New York, NY, USA, 2005. ACM.

[6] W. H. Dong, J. P. Liu, and Q. S. Guo. Generating effective schematic maps through generalization for mobilegis. In *the 4th International Symposium on LBS & TeleCartography*, 2007.

[7] M. Hampe, K. H. Anders, and M. Sester. Mrdb applications for data revision and real-time generalisation. In *Proceedings of the Proceedings of 21st International Cartographic Conference*, 2003.

[8] L. Harrie, L. T. Sarjakoski, and L. Lehto. A variable-scale map for small-display cartography. In *Proc. Symposium on GeoSpatial Theory, Processing, and Applications*, pages 8–12, 2002.

[9] Qingquan Li. Variable-scale representation of road networks on small mobile devices. *Comput. Geosci.*, 35(11):2185–2190, 2009.

[10] A. M. MacEachren. *How maps work*. Guilford Publications, 1995.

[11] M. Monmonier. *Mapping It Out*. The University Of Chicago Press, 1995.

[12] Manojit Sarkar and Marc H. Brown. Graphical fisheye views of graphs. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91, New York, NY, USA, 1992. ACM.

[13] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Softw.*, 11(6):70–77, 1994.

[14] R. Spence. Information visualization. *Addison-Wesley & ACM Press*, 2001.

Hang Yu obtained his B.Eng. and M.Eng. degrees in Refrigeration and Cryogenics Engineering from Shanghai JiaoTong University, China in 1998 and 2001. He obtained his Ph.D. degree in 2007 from the Department of Computer Science, National University of Singapore. The research topics were on image and media processing, volume graphics. Currently, he is a research fellow in the Institute for Infocomm Research, Singapore.

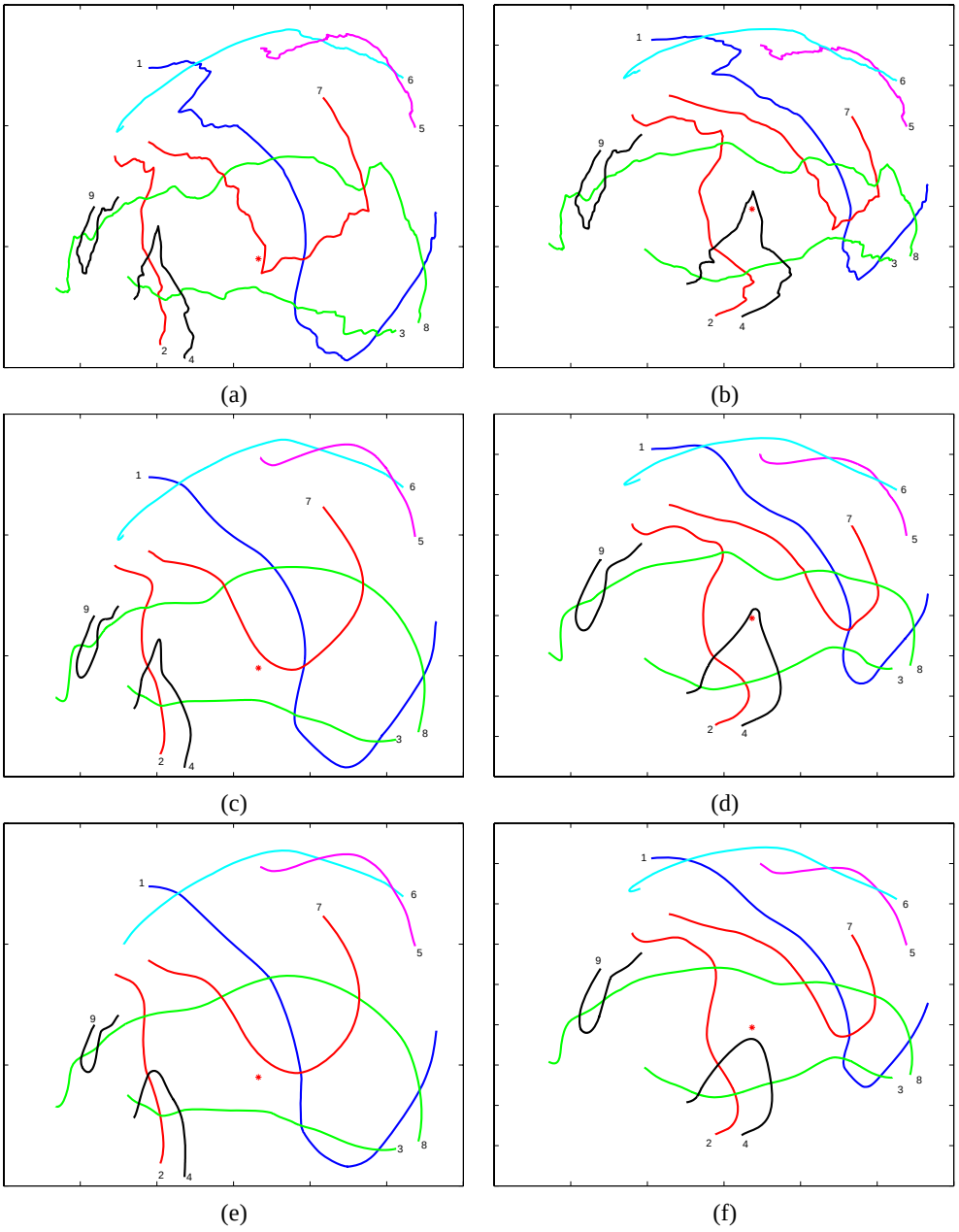


Fig. 6. Fisheye view transformation plus line smoothing on 9 polylines. Distortion factor d = 3. Area difference threshold γ = 0.01 for (c) and (d). γ = 0.3 for (e) and (f).

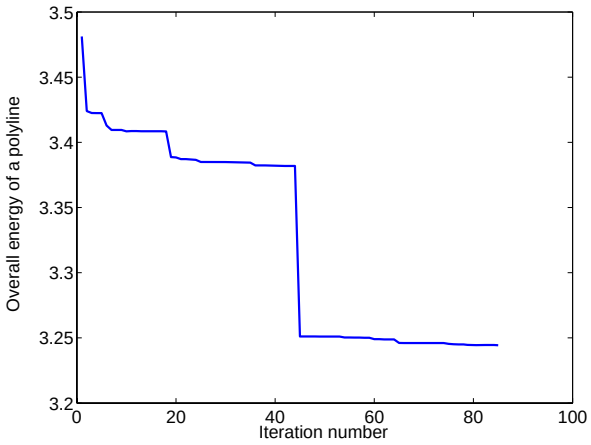


Fig. 7. Overall energy of a polyline 1 in Figure 6 (c) versus the iteration number.

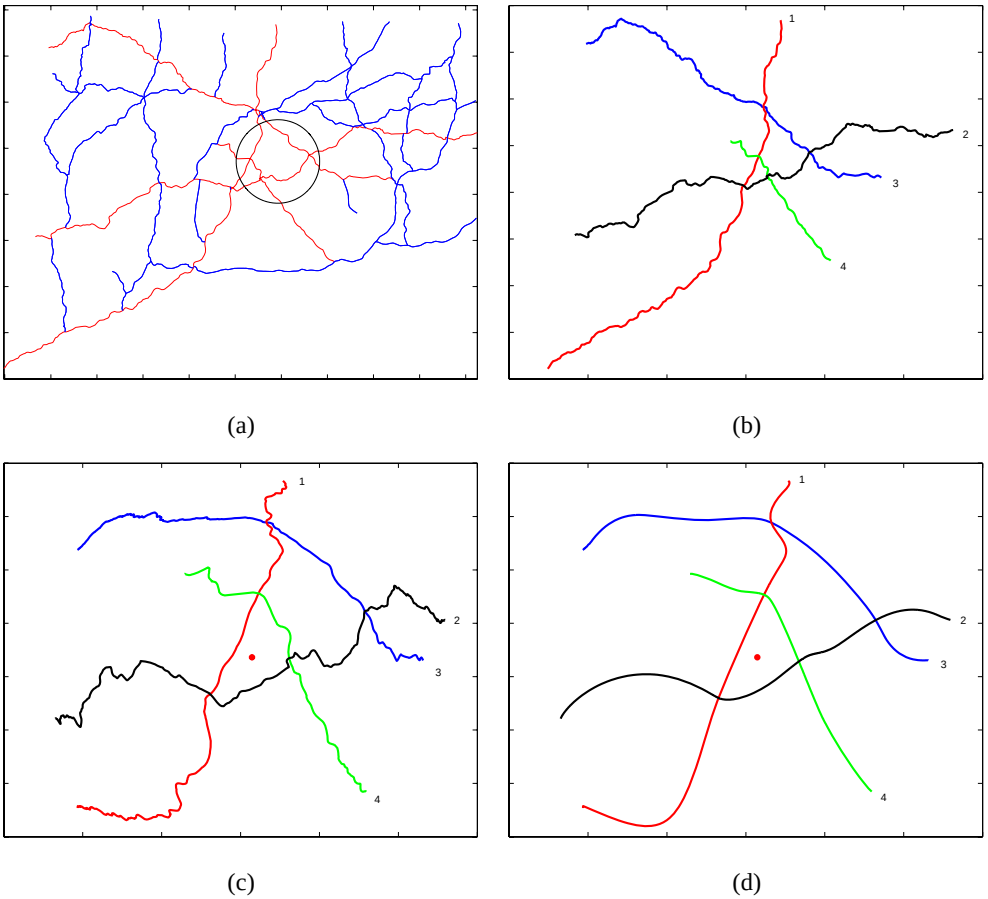


Fig. 8. The route map data-set is extracted from a simple representation of the major roads in the state of Connecticut, US. It depicts the highway network in the state at 1:250,000 scale. The black circle in (a) indicates the ROI with 4 routes depicted as red color. Fisheye view distortion factor d = 3 for (c). Area difference threshold γ = 0.01 for (d).