

A COMPLETE SOLUTION OF CARTOGRAPHIC DISPLACEMENT BASED ON ELASTIC BEAMS MODEL AND DELAUNAY TRIANGULATION

Y. G. Liu^{a,c,*}, Q. S. Guo^{a,b,*}, Y. G. Sun^a

^aSchool of Resource and Environment Science, Wuhan University, Wuhan 430079, China - liuygis@foxmail.com, guoqingsheng@126.com, 308569562@qq.com

^bState Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China - guoqingsheng@126.com

^cSchool of GeoScience, Yangtze University, Wuhan 430100, China - liuygis@foxmail.com

Commission IV, WG IV /2

KEY WORDS: Cartographic generalization, Displacement, Conflict, Elastic beams model, Delaunay triangulation, Skeleton, Proximity graph

ABSTRACT:

In map production and generalization, it is inevitable to arise some spatial conflicts, but the detection and resolution of these spatial conflicts still requires manual operation. It is become a bottleneck hindering the development of automated cartographic generalization. Displacement is the most useful contextual operator that is often used for resolving the conflicts arising between two or more map objects. Automated generalization researches have reported many approaches of displacement including sequential approaches and optimization approaches. As an excellent optimization approach on the basis of energy minimization principles, elastic beams model has been used in resolving displacement problem of roads and buildings for several times. However, to realize a complete displacement solution, techniques of conflict detection and spatial context analysis should be also take into consideration. So we proposed a complete solution of displacement based on the combined use of elastic beams model and constrained Delaunay triangulation (CDT) in this paper. The solution designed as a cyclic and iterative process containing two phases: detection phase and displacement phase. In detection phase, CDT of map is use to detect proximity conflicts, identify spatial relationships and structures, and construct auxiliary structure, so as to support the displacement phase on the basis of elastic beams. In addition, for the improvements of displacement algorithm, a method for adaptive parameters setting and a new iterative strategy are put forward. Finally, we implemented our solution on a testing map generalization platform, and successfully tested it against 2 hand-generated test datasets of roads and buildings respectively.

1. INTRUCTION

In map production and generalization, it is inevitable to arise some spatial conflicts where map objects overlap, or become too close together, then the spatial relationships between map objects cannot be understand correctly. Many contextual map generalization operators (e.g., selection, exaggeration, displacement, typification) can be used to resolve these conflicts. Among these operators, displacement is the most important operator that is often used for resolving the problems arising when two or more map objects are in conflict. During map generalization, the spatial conflicts pertaining to displacement arise from 3 reasons (Basaraner 2011): (1) the decrease of absolute empty space between two objects due to the scale reduction (2) the increase of symbol size, and (3) the change in the geometries of objects owing to previous generalisation algorithms. Typically, two kinds of displacement problems can be distinguished (Bader 2001): (1) deformation of linear map

objects including lines and bigger polygons' border lines, and (2) translation of rigid map objects including small polygons and points. Deformation of linear map objects is subject to high requirements regarding preservation of shape, while translation of rigid map objects needs consider constraints for maintaining spatial relationship and patterns. So far, most of cartographic displacement is still implemented in a manual mode, and it is become a bottleneck hindering the development of automated cartographic generalization.

Cartography and GIS community has reported many approaches of automated displacement including sequential approaches and optimization approaches (Guo Qingsheng et al. 2007). For sequential approaches, map objects are displaced one by one, step by step, there is no overall evaluation of the result. The most representative achievement of this category of approaches may be the equation for computing displacement vectors proposed by Lichtner (Lichtner 1979). Different with sequential

* LIU Yuangang(1982—), PhD candidate, majors in cartographic generalization.

E-Mail: liuygis@foxmail.com

* GUO Qingsheng(1965—), male, PhD, professor, majors in cartographic generalization, intelligent handling and visualization of geographical information

E-mail: guoqingsheng@126.com

approaches, optimization approaches take the geometries of all map objects into account, and resolve all of the conflicts simultaneously, which can be further divided into combinatorial optimization approaches and function optimization approaches. In combinatorial optimization approaches, the new locations of map objects are calculated using trials, and the trials can be based on heuristic search algorithms, such as simulated annealing (Ware et al. 2003) and genetic algorithm (Wilson et al. 2003). In function optimization approaches, the displacement problem of map objects is modelled as a global equation system based on some methods developed in mathematics, physics and/or engineering, such as energy minimization methods (Burghardt and Meier 1997; Bader 2001), finite element method (EFM) (Højholt 2000) and least squares adjustment (Harries 1999, Sester 2005).

As an optimization approach based on energy minimization principles, the displacement algorithm based on elastic beams resolves the displacement problem through an interplay of external forces, pushing the conflicting objects towards admissible places of interest, and internal regularization forces, attempting to make the shape or structure of objects resemble its initial form (Bader 2001). Previous studies have illustrated that the elastic beams model could be successfully applied to resolve the displacement problems of roads and buildings (Bader 2001, Bader et al. 2005). However, to implement an automated displacement solution based on elastic beams model, besides the model itself, there are still some other problems need to be considered. At first, we should provide feasible techniques to identify contextual environment and detect conflicts. But, unfortunately, most existing commercial mapping and GIS software have not provided available tools to support this preparative phase of displacement. Second, the application of displacement algorithm based on elastic beams is not as perfect as we wished. There are still some aspects needs to be improved, such as setting of its parameters and iterative strategy.

This paper aims to propose a complete solution of displacement based on the combined use of elastic beams model and constrained Delaunay triangulation (CDT). The solution designed as a cyclic and iterative process containing two phases: detection phase and displacement phase. In detection phase, CDT of map is use to detect proximity conflicts, identify spatial relationships and structures as well as construct auxiliary structure, so as to support the displacement phase on the basis of elastic beams. Furthermore, as important components of the overall solution, a method for adaptive parameters setting and a new iterative strategy are put forward to improve the displacement algorithm. Finally, to empirically validate the proposed solution, we implemented a displacement function modules on a testing map generalization platform, and tested our solution against 2 hand-generated test datasets of roads and buildings respectively.

2. METHODS

From the existing approaches we can learn that the key issues of displacement are: where are the spatial conflicts? how to resolve the conflict through displacing and deforming the involved map objects? It is mean that a complete displacement solution at least contains two phases: detection phase and displacement phase. Therefore, our displacement solution is designed as a cyclic and iterative process including contextual information identification, conflicts detection and forces calculation, and displacement.

2.1 Contextual Information Identification and Representation

2.1.1 Constrained Delaunay triangulation: Delaunay triangulation (DT) is one of the most powerful tool for proximity modelling and analysis. The introduction of constraints to create a constrained Delaunay triangulation (CDT) results in the loss of the empty circumcircle criterion, but the triangulation is still the most equiangular triangulation, given the constraints (Jones, et al., 1999). Here, we adopt CDT to identify and represent the characteristics, relationships and structures of map objects, where line segments of map objects (e.g., buildings and roads) are constrained edges. The construction of CDT has been researched in the Computational Geometry community for many years. So we will concentrate on the process and principles of identification of spatial relationships and structures, but not on the CDT algorithms. However, what need to emphasize is that the CDT should be refined by inserting extra points to some long constrained lines to avoid generating very narrow triangles. Otherwise, the CDT may not correctly reflect the proximity relationships (Ai Tinghua and Guo Renzhong, 2000). The interval between inserted points can be based on the statistics of the line segments (e.g., the minimal length of the line segments).

2.1.2 Skeleton extracting based on CDT: A skeleton extracting algorithm based on CDT is implemented and adopted to extracting the skeleton structure of the gap space between map objects. In cartographic generalization, the skeleton structure based on CDT had been applied in identification, characterization, classification of proximity conflicts for road network (Thom 2007). In our displacement solution, the uses of the derived skeleton structure are twofold. On the one hand, a kind of dual graph of the skeleton structure, proximity graph, can be easily constructed, so as to provide an auxiliary structure for the displacement of rigid map objects. On the other hand, with the proximity relationships information provided by the skeleton, the detection procedure for checking whether a pair of neighbouring objects is in conflict or not can be simplified by measuring the minimal distance of the gap space.

In the CDT, each gap space between a pair of map objects is covered by a sequence of triangles, which is termed a path. Given a path, its axis can be extracted, which is named skeleton arc here. Extracting and connecting all the skeleton arcs and their end nodes, a skeleton structure of the map gap space can be constructed. The main idea of the algorithm is: (1) Trace the triangle paths in the CDT, (2) Generate skeleton arcs by extracting the axes of these paths, and (3) Using derived skeleton arcs and their end nodes, construct the skeleton structure through topological organization method. To trace the triangle paths in CDT, the triangles need to be divided into 4 types (figure 1). They are branching triangles, trunk triangles, leaf triangles, and isolated triangles, respectively refer to those having no constrained edge, those having only one constrained edge, those having two constrained edges, and those having three constrained edges. The isolated triangles are out of our consideration for their specificity. During the path tracing process, branching triangles and leaf triangles are used as entrances or the exits of the paths, while trunk triangles are used as middle parts of the paths. When each leaf triangle had been traced one time and each branching triangle had been traced three times, all the paths could be pick out. At the meantime, the corresponding skeleton arcs could be extracted through linking the skeleton segment in each triangles through the following means (Ai Tinghua 2004):

- Branching triangle: $O \rightarrow P_i$ or $P_i \rightarrow O$, $i = 1, 2, 3$
- Trunk triangle: $P_1 \rightarrow P_2$ or $P_2 \rightarrow P_1$
- Leaf triangle: $A_1 \rightarrow P_1$ or $A_1 \rightarrow P_1$

For details see figure 1, where P_1, P_2, P_3 is the midpoint of corresponding triangle edge, O is the centre point of the triangle, and V_1 is the vertex corresponding to P_1 .

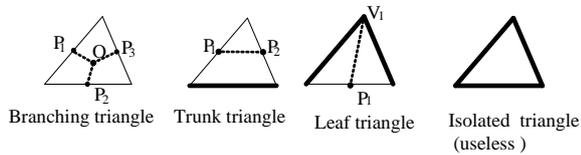


Figure 1. Four triangle types and their skeleton segments

In addition, two special cases should be considered. First, if a trunk triangle contains a non-constrained edge on the boundary of CDT, or contains a non-constrained edge having a node coincided with an end node of a line object, it should also be used as an entrance or an exit of path. Second, the derived skeleton arc set contains needless skeleton arcs in boundary polygon and concave areas of the map object. Since we only need consider the gap space between different objects, these needless skeleton arcs should be removed, and the connected nodes and arcs in the skeleton structure should be re-organized.

To represent the proximity information in the skeleton structure and support the following detection process, a class named *Skeleton_Arc* is defined:

```
class Skeleton_Arc
{
    MapObject LObj, RObj;
    List<Triangle> TList;
    List<Point> PList;
}
```

Fields *LObj* and *RObj* represent the two neighbouring objects on the left and right sides, field *TList* represents the sequence of triangles crossed by the skeleton arc, and field *PList* represents the coordinates of the skeleton arc.

2.1.3 Proximity graph construction: For the displacement of rigid map objects, auxiliary structure should be constructed to help the maintenance of the spatial relationships and structures, such as minimum spanning tree (MST) on the building centre point used in (Bader 2005). Here, a kind of dual graph of the skeleton graph, proximity graph, can be constructed and used as the auxiliary structure. In the proximity graph, each node represents a map objects and each edge represents a proximity relationship between a pair of map objects.

The proximity graph is a kind of abstract description of the proximity relationships of map objects. These relationships can be classified into three types, including relationship between a pair of rigid objects, relationship between a rigid object and a linear object, and relationship between a pair of linear objects. We just use the auxiliary structure to support the displacement of rigid objects, so only the former two types of relationship need to be considered. To represent the spatial relationships and structures of a map partition, the nodes of constructed proximity graph should be located at some representative points of map objects. For the two kinds of map objects involving, rigid

objects and linear objects, different cases are considered. A node of a rigid object is located at the centre point of its border polygon, whereas a node on a linear object is located at the nearest point to the corresponding neighbouring node of rigid object (figure 5(b)).

2.2 Conflicts Detection and Forces Calculation

In the process of cartographic generalization, displacement is triggered by proximity conflict which is a kind of conflict generated because of no enough white area between two neighbouring objects. Therefore, to detect proximity conflicts, the gap space between neighbouring objects should be identified in advance. The skeleton arc is a kind of axis line representing the equidistance to the objects on both sides. Using the skeleton arcs described in section 2.1.2, we can easily check whether a conflict arises between a pair of neighbouring objects and measure the severity of the conflict. To provide conditions for the following displacement based on elastic beams, here, the severities of detected conflicts are quantified as push forces on those conflicted objects.

2.2.1 Conflicts detection: To check whether a pair of neighbouring objects is in conflict, the minimal distance of the two objects should be calculated. Then compare the minimal distance with a threshold, if it is smaller than the threshold value, the two objects are in conflict. In each triangle in the triangle sequence of a skeleton arc, a local minimal distance of the corresponding two objects can be calculated. Calculating and comparing of these local minimal distance in each triangle one by one, the minimal distance and the nearest two points of the two neighbouring objects can be found out. Usually, the minimal distance threshold is determined according to graphic limits (e.g., 0.2mm). Considering the sizes of map symbols, the threshold value can be given by:

$$d = d_{\min} + \frac{1}{2}(r_1 + r_2) \quad (1)$$

Where d_{\min} is the minimal distance threshold, r_1 and r_2 are the symbol sizes of the two objects. If the minimal distance between the two object less than d , they are in conflict.

One thing need to note is that when detecting the conflict between a pair of linear objects, areas around junctions should be excluded (Bader 2001, Thom 2007). It is because of that two linear objects sharing a common node always have distance 0, but it should not be detected as a conflict. The radius of the exclusion scope around a junction can be set as the corresponding minimal distance threshold, d .

For the different natures of the two displacement operations (i.e. deformation of linear objects and translation of rigid objects), two types of conflicts, linear conflicts and rigid conflicts, are distinguished. Linear conflicts refer to those arise between a pair of linear objects, and rigid conflicts refer to those arise between a pair of rigid objects or between a rigid object and a linear object. A Linear conflict is represented as a conflict area by tracing a continuous bottleneck area between the two conflicted linear objects through the skeleton arc (figure 3(c)), where the width is less than the predefined tolerance d . A rigid conflict is represented as a conflict line by connecting the nearest two points on the two conflicted objects (figure 5(c)).

2.2.2 From conflicts to individual external forces: In the elastic beams model, displacement is interpreted as an optimization problem rely on an interplay of external and internal forces (Bader 2001, Bader et al. 2005). Proximity conflicts give rise to external forces, which is the primary trigger for the displacement. In order to provide input for elastic beams model, we first calculated the individual external forces caused by the detected proximity conflicts, and then combine them on each rigid object or each vertex of linear objects into resultant forces. On account of the different types of conflicts, the calculation methods are distinguished.

For linear conflict, the individual force acts on each conflicted vertex can be calculated as follows:

$$f = \left(\frac{d}{2} - \min(|\mathbf{v}|, \frac{d}{2})\right) \frac{\mathbf{v}}{|\mathbf{v}|} \quad (2)$$

Where \mathbf{v} is the vector from the closed point on skeleton arc to the vertex, d is the distance threshold, and f is the calculated force. For rigid conflict, two cases are distinguished. If it is a conflict between two rigid objects, the individual forces act on the left and right ones are calculated respectively as follows:

$$f_l = \frac{A_r}{A_l + A_r} * (d - \overline{P_l P_l}) \frac{\overline{P_l P_l}}{|\overline{P_l P_l}|} \quad (3)$$

$$f_r = \frac{A_l}{A_l + A_r} * (d - \overline{P_r P_r}) \frac{\overline{P_r P_r}}{|\overline{P_r P_r}|} \quad (4)$$

Where A_l and A_r are the areas of the left and right objects respectively, P_l and P_r are the two points with the minimal distance between left and right objects, d is the distance threshold, and f_l and f_r are the calculated forces on the left and right objects respectively. If it is a conflict between a rigid object and a linear object, the linear object is set as an immovable boundary line of the map partition according to usual practice, so the individual force act on the rigid object is calculated as follows:

$$f = (d - \overline{P_L P_R}) \frac{\overline{P_L P_R}}{|\overline{P_L P_R}|} \quad (5)$$

Where P_L and P_R are the two nearest points on the linear object and rigid object respectively, d is the distance threshold, and f is the calculated force.

2.2.3 Resultant force calculation: When a rigid object or a vertex of linear object is acted upon by more than one external forces, we should combined them into resultant forces. Calculating the resultant force based on the simple vector-based sum approach may cause inappropriate displacement. Hence, a more reasonable method to combine the individual forces is proposed. first, using the direction of the maximal force as the direction of horizontal axis, we establish a local coordinate system. Then we resolve each individual force along the local coordinate axes. Finally, we can the appropriate resultant force by summing the single largest component in each of the four main directions. Figure 2 demonstrates this for the individual forces F_1 , F_2 and F_3 . For more examples of calculated resultant forces see figure 3(d) and figure 5(d).

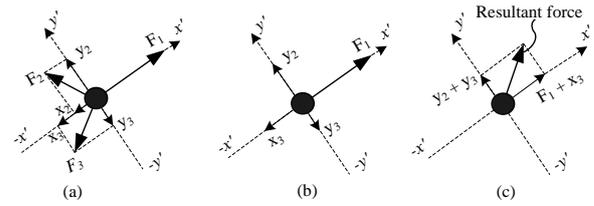


Figure 2. Calculating Resultant force: (a) Establish a local coordinate using direction of the maximal force (F_1) as the x' axis direction, and resolve each individual force along the local coordinate axes (x' , $-x'$, y' , $-y'$), (b) Select the largest components, (F_1 , x_3 , y_2 , y_3), in each of the four main directions, and (c) Calculate the resultant force.

2.3 Displacement Based on Elastic Beams Model

Two improvements of the displacement algorithm based on elastic beams model are put forward here. Detailed description of elastic beams model can be found in (Bader 2001).

2.3.1 Adaptive setting of parameters : The displacement result affected by parameters of the elastic beams model, i.e., elastic modulus (E), moment of inertia (I), and cross-sectional area (A). In order to reduce the complexity of influence factors as well as optimize the setting of parameters, parameters I and A are considered as constants, but the value of E is varied to scale the displacement magnitude. Under these conditions, we proposed a method to estimate the appropriate value of E through calculating the displacement twice. In the first calculation, E is assigned by an arbitrary value E_0 (>0), then the initial displacement is calculated. Assuming the maximal value of displacement is always corresponding to the maximal value of forces, the appropriate value of E can be estimate through an empirical formula as follows:

$$E = \frac{d_{\max}^0}{f_{\max}} \cdot E_0 \quad (6)$$

Where f_{\max} is the maximal value of forces, d_{\max}^0 is the displacement value corresponding to f_{\max} in the first calculation, E_0 is the initial value of E . In the second calculation, using the estimated value of E , the new maximal value of displacement d_{\max} can be made to almost equal to the maximal value of forces ($d_{\max} \approx f_{\max}$). Thus the most serious conflict can just be solved.

2.3.2 Iterative strategy : It is usually cannot guarantee that all conflicts will be resolved by calling the displacement algorithm only once. Hence, the workflow of complete solution should be designed as a cyclic and iterative process including repeated conflicts and contextual information identification and resolution. Here, we adopt a different iterative strategy with (Bader 2001), which can be described in matrix terms by

$$\mathbf{K}^{(t-1)} \mathbf{d}^{(t)} = \mathbf{f}^{(t-1)} \quad (7)$$

Where \mathbf{K} , \mathbf{d} and \mathbf{f} is stiffness matrix, displacement vector and force vector respectively, t is iteration time. In each iteration, not only the force vector but also the stiffness matrix are recalculated according to the immediate state. In this way, we make the external forces dominate the inner ones during the iterations, and it is more likely to resolve all proximity conflicts.

3. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed solution and algorithms were implemented as function modules in our testing generalization platform which has been developed based on the ArcGIS Engine developer kit. On the testing platform we implemented CDT, skeleton, proximity graph, and elastic beams displacement algorithms in C#. We tested our solution against 2 hand-generated test datasets of roads and buildings respectively. In each dataset, there is only one map partition. In the experiments, the parameters of elastic beams model I and A are set as constants, but parameter E is adaptively determined in each iterations. According to our experiences, results of roads displacement are more reasonable when $I > A$ (e.g., $I=10^7$, $A=1$), whereas for building displacement, the setting of $I \approx A$ (e.g., $I=1$, $A=1$) is more appropriate. The minimum distance threshold between map objects is 0.2mm.

Road dataset (scale 1:1000k), contains 3 classes of roads including national road (1.2mm), provincial roads (0.9mm), and county road (0.9mm), described by 210 vertexes. The roads are represented by double line symbols. Figure 3 demonstrates the intermediate states of CDT construction, skeleton extraction, conflict detection and force calculation in the 1st iteration. After 7 iterations, the severity of the conflict had been reduced to an acceptable level. Figure 4 shows the final result and the comparison of the initial and the final state. On the result map, all the proximity conflicts are resolved, and there is only a little change in the shapes of the displaced roads.

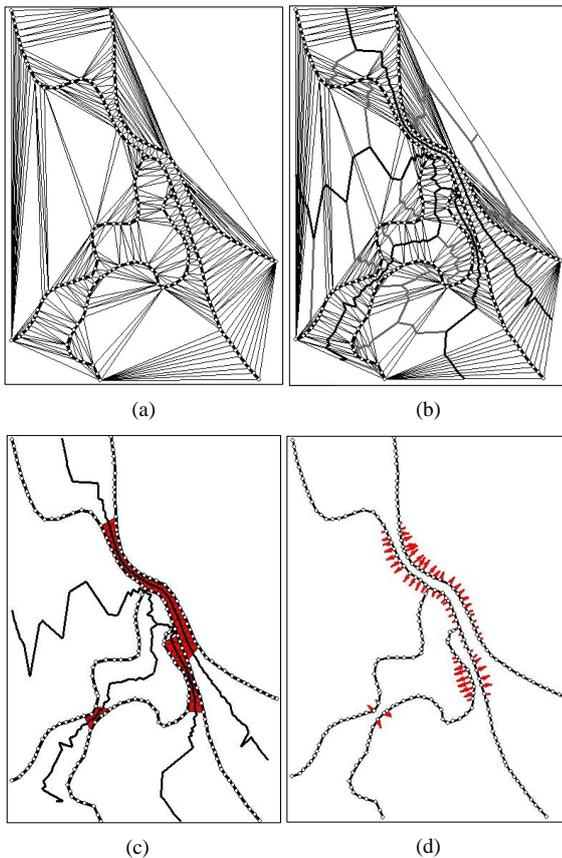


Figure 3. Process of conflict detection and forces calculation: (a) Constructing CDT, (b) Extracting skeleton, (c) Detecting conflicts, and (d) Calculating forces (enlarged 3 times).

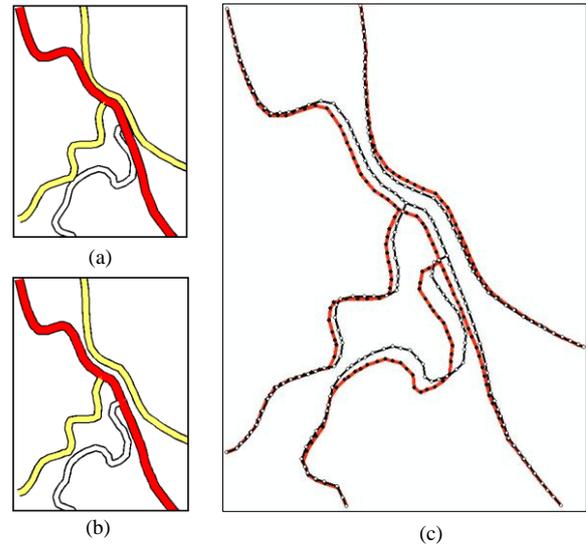


Figure 4. Result of displacement: (a) Initial roads (1:1000k), (b) Displaced road (1:1000k), and (c) Comparing the Initial and final state

Building dataset (scale 1:10k), contains 27 buildings in a street block. On the map, buildings are represented by area symbols with solid outline (0.1mm), streets are represented by double line symbols (1.2mm). Figure 5 demonstrates the intermediate states of CDT construction, skeleton extraction, proximity graph construction, conflict detection and force calculation in the 1st iteration. After 8 times iterations, all of the conflicts were resolved. Figure 6 shows the final result and the comparison of the initial and the final state. We can find that the result is much more legible than the original map, and the spatial relationships and structures of the building block are kept very well.

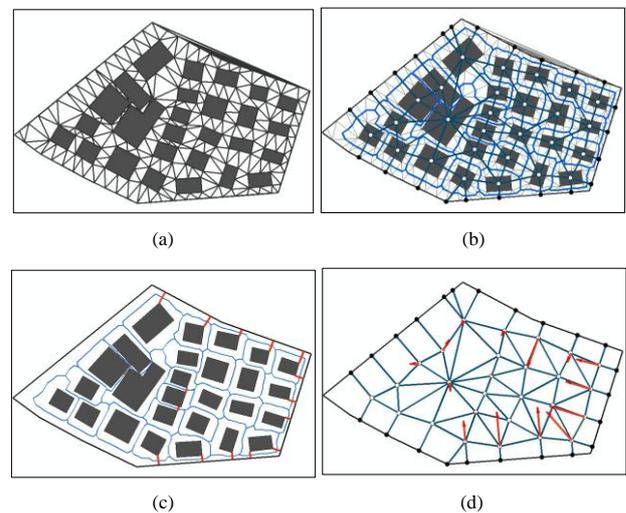


Figure 5. Process of proximity graph construction, conflict detection and forces calculation: (a) Constructing CDT, (b) Extracting skeleton and constructing proximity graph, (c) Detecting conflicts, and (d) Calculating forces (enlarged 5 times).

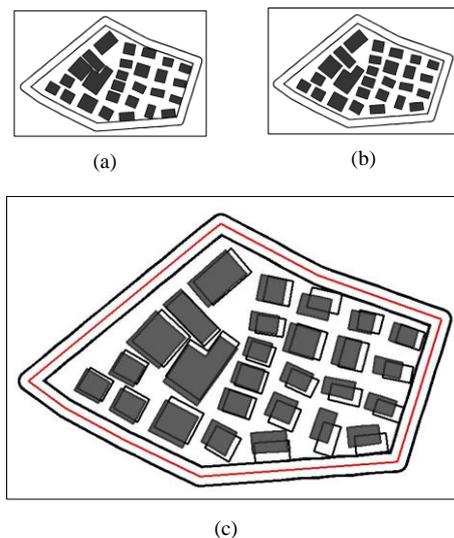


Figure 6. Result of displacement: (a) Initial buildings (1:10k), (b) Displaced buildings (1:10k), and (c) Comparing the Initial and final state.

4. DISCUSSION AND CONCLUSIONS

The experimental results show that the solution is feasible. Based on CDT, several models and algorithms (i.e., skeleton extraction, proximity graph construction, conflicts detection, forces calculation) are implemented to support the preparatory process for the displacement. Using displacement algorithm of elastic beams, the displacement and its propagation can be realized simultaneously with good preservation of spatial relationships and structures. Through the physical concept, force, the two techniques are integrated together to provide a complete solution of displacement. In this way, the advantages of the two techniques are effectively used in the two types of displacement problems, deformation of linear map objects and translation of rigid map objects. Comparing with the work of (Bader 2001), we made two improvements to the displacement algorithm based on elastic beams: (1) A method to adaptively estimate the appropriate parameters of elastic beams model, and (2) A different iterative strategy, by which the external forces can be made to dominate the inner ones, so the proximity conflicts are more likely to be resolved.

However, the reliability of the improved displacement algorithm is based on whether the maximum value of displacement is corresponding to the maximum value of forces. If the condition is satisfied, we can get perfect results. Otherwise, it may cause too much distortion or shift to some objects or vertices that have relative small forces. This may be the main drawback of the proposed solution. So for more successful displacement, the positional accuracy constraint should be explicitly considered to prevent displacing map objects too far from it original positions.

ACKNOWLEDGEMENTS

This research is supported by National Natural Science Foundation of China, No. 41071289, No. 41171350, No. 41101351; and the Fundamental Research Funds for the Central Universities, No. 2012205020212.

REFERENCES

- Ai, T. H., 2004. A displacement of building cluster based on field analysis. *Acta Geodaetica et Cartographica Sinica*, 33(1), pp. 89-94.
- Ai, T.H. and Guo, R. Z., 2000. Extracting center-lines and building street network based on constrained Delaunay triangulation. *Acta Geodaetica et Cartographica Sinica*, 29(4), pp. 348-354.
- Bader, M., Barrault, M., Weibel, R., 2005. Building displacement over a ductile truss. *International Journal of Geographical Information Science*, 19 (8-9), pp. 915-936.
- Bander, M., 2001. *Energy Minimization Methods for Feature Displacement in Map Generalization*. Department of Geography, University of Zurich, Zurich.
- Basaraner, M., 2011. A zone-based iterative building displacement method through the collective use of Voronoi tessellation, spatial analysis and multicriteria decision making. *Bol. Ci ênc. Geod. (Online)*, 17(2) pp. 161-187.
- Burghardt D. and Meier S., 1997. Cartographic displacement using the snakes concept. In: W. Foerstner and L. Pluemer (editors), *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*. Birkhaeuser-Verlag, Basel.
- Guo, Q. S., Huang, Y. L., Zheng, C. Y., Cai Y. X., 2007. *Spatial Reasoning and Progresssive Map Generalization*. Wuhan University Press, Wuhan, pp. 151-153.
- Harrie, L., 1999. The constraint method for solving spatial conflicts in cartographic generalization. *Cartography and Geographical Information Science*, 26(1), pp. 55-69.
- Højholt, P., 2000. Solving space conflicts in map generalization: using a finite element method. *Cartography and Geographic Information Science*, 27 (1), pp. 65-74.
- Jones, C. B., Ware, J. M., Eynon C.D., 1999. Triangulated spatial models and neighbourhood search: an experimental comparison with quadtrees. *The Visual Computer*, 15, pp. 235–248.
- Lichtner W., 1979. Computer-assisted processes of cartographic generalization in topographic maps. *Geo-Processing*, 1(1), pp. 183-199.
- Sester, M., 2005. Optimizing approaches for generalization and data abstraction. *International Journal of Geographic Information Science*, 19(8-9), pp. 871-897.
- Thom, S., 2007. Automatic resolution of road network conflicts using displacement algorithms orchestrated by geographical agents. In: *10th ICA Workshop on Generalisation and Multiple Representation*, Moscow.
- Ware, J. M., Jones, C. B., Thomas, N., 2003. Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8), pp. 743–769.
- Wilson, I. D., Ware, J. M., Ware, J. A., 2003. A genetic algorithm approach to cartographic map generalization. *Computers in Industry*, 52 (3), pp. 291–304.