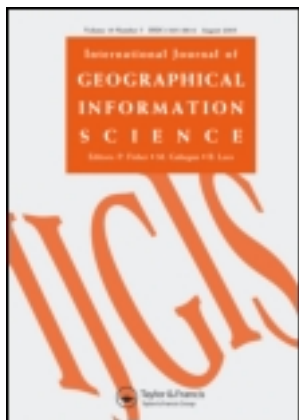


This article was downloaded by: [Institute of Remote Sensing Application]
On: 08 November 2011, At: 23:29
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered
office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Geographical Information Science

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/tgis20>

Interactive visualization of multi-resolution urban building models considering spatial cognition

Ling Yang^a, Liqiang Zhang^a, Jingtao Ma^b, Jinghan Xie^a & Liu Liu^a

^a State Key Laboratory of Remote Sensing Science, Jointly Sponsored by Beijing Normal University and the Institute of Remote Sensing Applications of Chinese Academy of Sciences, Beijing, PR China

^b PTV America, Inc., Tacoma, WA, USA

Available online: 18 Feb 2011

To cite this article: Ling Yang, Liqiang Zhang, Jingtao Ma, Jinghan Xie & Liu Liu (2011): Interactive visualization of multi-resolution urban building models considering spatial cognition, International Journal of Geographical Information Science, 25:1, 5-24

To link to this article: <http://dx.doi.org/10.1080/13658816.2010.488239>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.tandfonline.com/page/terms-and-conditions>

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Interactive visualization of multi-resolution urban building models considering spatial cognition

Ling Yang^a, Liqiang Zhang^{a*}, Jingtao Ma^b, Jinghan Xie^a and Liu Liu^a

^aState Key Laboratory of Remote Sensing Science, Jointly Sponsored by Beijing Normal University and the Institute of Remote Sensing Applications of Chinese Academy of Sciences, Beijing, PR China;
^bPTV America, Inc., Tacoma, WA, USA

(Received 8 October 2009; final version received 15 April 2010)

Multi-resolution visualization of massive urban buildings is one of the most important components for a cyber city. Urban is a highly humanized system, and thus visualizing urban buildings needs to abide by people's habits of cognizing spatial relations between objects for their accurate and quick understanding of urban spatial information. This article proposes an approach to generalize and render urban building models in the context of Gestalt psychology and urban legibility. We introduce a new distance measurement method as the distance metric for the single-link clustering algorithm, which is used to group building footprints into clusters. Each cluster is merged based on the Delaunay triangulation and the polyline generalization algorithm. We then construct a hierarchical tree to store multi-resolution building models and implement interactive three-dimensional visualization of large-scale and high-density urban buildings. Experimental results indicate that the proposed methodology not only reduces the geometric complexity of urban models but also preserves urban legibility successfully and follow Gestalt principles.

Keywords: massive urban models; LOD; hierarchical tree; spatial cognition

1. Introduction

An urban resembles the layouts, activities and functionalities of a real-world community. Thus, urban models are essential components to help people comprehend spatial and temporal relationships. The three-dimensional (3D) urban visualization is experiencing a fast-growing body of research and development efforts from many scientists. Because of the great complexity of urban scenery, which combines large extents with very rich small-scale visual details, real-time visualization of virtual cities is understandably a challenging problem. Building up level-of-detail (LOD) models can significantly assist interactive rendering of massive urban building models. Therefore, it becomes necessary to apply principles of generalization to their visualization to communicate spatial information efficiently. Because urban is a highly humanized system, the geometric relations of the models and human spatial cognition should be considered in generalization and visualization of massive building models. On the premise of keeping spatial geometrical precision at a certain level, visualization of large building models, which fits people's vision habit and

*Corresponding author. Email: zihaozhang2003@yahoo.com.cn

urban legibility, can help people gain a comprehensive view of the structures and activity occurring at highly occluded urban settings.

Some traditional mesh generalization methods were proposed to simplify single complex objects (e.g. terrains). However, their effects are limited when these approaches are applied to simplify large-scale urban scene, because a city often contains a large number of buildings and the majority of buildings are already low-polygon objects. Others mostly focus on discrete LOD where buildings beyond a certain visual distance are not rendered.

Because geometry is highly discontinuous and different views of the models have widely different depth complexity that ranges from full visibility of flyovers to nearly full occlusion at ground level, the above approaches do not fit well within urban models rendering (Cignoni *et al.* 2007). In this article, 2.5D building models are grouped and generalized based on Gestalt psychology and urban legibility. The proposed approach first performs single-link clustering of building models based on a new distance criterion; then merges the clustering results using Delaunay triangulation and the polyline generalization algorithm in real time; finally, the approach automatically generates multi-resolution building models, and interactive 3D visualization of large-scale and high-density urban buildings is achieved.

2. Related work

2.1. Two-dimensional building footprint generalization

Although automated building simplification is a well-studied area of cartography, most approaches focus on simplifying 2D building footprints (Sester 2000). The 2D map generalization methods, such as artificial neural networks (Allouche and Moulin 2005) and least squares adjustment (Harrie 1999, Sester 2000), have been applied to yield generalization operators for automatically deriving small-scale maps. Brassel and Weibel (1988) proposed a knowledge-based algorithm, which is widely referenced and used. The method of Kada and Luo (2006) can reduce the complexity of 2D ground footprints by defining parts of simplified buildings as intersections of half-planes and by cell decomposition. Although this approach retains the overall appearance of the original building ground plans, it sometimes yields erroneous, self-intersecting lines. Meanwhile, the relations between buildings were detected and organized (Regnauld 1996, 2001) by using the minimum spanning tree. Regnauld then used a graph of proximity to segment the building set based on certain criteria taken from Gestalt theory. Li *et al.* (2004) described an approach for building groupings and generalizations. They adopted the concepts developed in urban morphology for the global organization of building groups and the formation of hierarchies. In their approach, Gestalt principles are employed as local constraints to further group enclaves, blocks, superblocks or neighbourhoods. Such analysis provides the semantic information associated with geometrical objects of buildings to be generalized. However, the methodology they presented is difficult to deal with in U-type buildings and other buildings with complex shapes.

2.2. Three-dimensional city models generalization

Compared to 2D map generalization techniques, generalization in 3D spatial objects is still in its infancy (Glander and Döllner 2009). Many algorithms have been proposed (Li 2007), which employed different strategies to eliminate minor parts of a building and small features. This leads to parameters with different semantics, which cause different effects on building generalization. More recent techniques for simplifying buildings in 3D are designed to find

and eliminate small volumetric features like protrusions on the surface model (Thiemann 2002, Forberg 2007). Thiemann (2002) divided the whole generalization process into a segmentation, interpretation and generalization procedure. In the segmentation step, a boundary representation's building was segmented into a set of convex parts that were stored as a cell complex in a Constructive Solid Geometry (CSG) tree. The CSG tree presents a hierarchical subdivision; however, as such it is not exactly a generalization. For 3D visualization systems, especially web-based visualization systems, it is better to avoid adding new vertices. Royan *et al.* (2003, 2006) proposed a Delaunay triangulation-based merging algorithm and a binary tree structure called PBTree, to represent densely urban areas over the network. His approach allows users to navigate freely in the city scene. However, the generated PBTree has many levels. Meng and Forberg (2007) gave an overview of 3D generalization issues. They attempted to provide a basis from which 2D operations, such as aggregation, typification and landmark exaggeration (Bai and Chen 2001), can be extended to cope with 3D generalization problems. Anders (2005) proposed an algorithm for simplifying 3D urban models by aggregating nearby buildings. He projected the building models onto three orthogonal planes and obtained simplified models based on the projections. However, it is only suitable for simple symmetric models that do not self-occlude during the projections. A scale-space approach was introduced to generate LOD representations of 3D city models (Forberg and Mayer 2002, Forberg 2007). Because this approach is only employed to deal with orthogonal building structures, it works by moving parallel facets towards each other until the facets meet and merge. Unfortunately, the squaring of non-orthogonal structures such as the treatment of roofs and walls is not accomplished.

CityGML was realized as an open data model and XML-based format for the storage and exchange of virtual 3D city models (Gröger *et al.* 2006). It allows storing building geometry in four levels of detail. This is limited to generalization regarding single buildings and therefore does not cover large changes of scale (Glander and Döllner 2009). Fan *et al.* (2009) presented an approach for building generalization by CityGML that considered the semantic information associated with geometrical meshes of the generalized buildings. Their approach can reduce storage space and speed up network transmission and geometric computation.

Compared to 2D maps, 3D scene simulates the real world directly and fits with people's spatial awareness better. Thus, 3D buildings' generalization should take into account the spatial psychology, and thus has more problems to be resolved. Glander and Döllner (2008) proposed a given infrastructure network to select landmark buildings in small scales and created an interactive visualization dynamically highlighting landmark buildings. They used a single building block model to replace individual models and dynamically exaggerated global landmark objects. Therefore, borders of single buildings are invisible to the observers, while preserving local landmarks in their representation, if needed. However, the landmarks are exaggerated, which often prevents users from making wrong distance estimations. An improved method was presented later by Glander and Döllner (2009) to automatically generalize hierarchical 3D city building models through creating several representations of increasing levels of abstraction. Using the infrastructure network, they grouped building models and replaced them with cell blocks. Chang *et al.* (2008) proposed an approach based on urban legibility theory. They used the single-link hierarchy method to generate building clusters. The landmarks were preserved by real-time computing pixel errors.

Based on Chang *et al.*'s (2008) work, we proposed an approach to make the generalization have a better coincidence with human special cognition. Our approach is divided into four steps: footprints clustering, clusters merging, generalization and LOD visualization.

3. A clustering method based on the theories of Gestalt psychology and urban legibility

According to the theory of Gestalt psychology including proximity, similarity, continuity and common orientation (Wertheimer 1923), separated elements in an image which have certain relations tend to be merged into a whole in people's mind. To obtain a good generalization representation of a city that abides by these criteria, the spatial relations, including distance, orientation, similarity and continuity, should be taken into account when building models are clustered. The key to build an excellent integration algorithm is how to obtain and make use of buildings' morphological features and spatial relations.

When performing the clustering and merging of building models, we consider not only the relations among adjacent buildings but also the urban-scale morphological characteristics. For 2D projection polygons (called footprints) of 2.5D building models, urban legibility (Lynch 1960) is applied to the clustering and merging processes. Urban legibility refers to highly generalized images of urban environments in human minds. It gives five kinds of elements forming a city's mental image: roads, edges, regions, nodes and landmarks.

Roads are important features of urban legibility. Edges refer to the linear entities such as rivers. Regions refer to urban's different function areas, such as central business districts. Nodes can be linking points, such as road crossings, and also be certain aggregation points, such as squares. Landmarks are recognizable objects that have distinctive figures, for example, those buildings which are much higher than surrounding objects, and distinguishable objects used for orientation. While users navigate in virtual urban landscapes, these five major elements can help them identify spatial locations effectively. Therefore, they should be maintained during generalizations to make the simplified urban environments fit the urban legibility in human minds. Among the five elements, roads are the most important features of urban legibility (Lynch 1960) and should be of priority in preservation. Tower buildings can be easily identified. Other elements, such as regions and nodes, are also considered to a certain extent.

3.1. Generation of hierarchical clusters

There are two aims of clustering. The first one is to identify the distribution pattern of urban buildings. The second one is to improve computation efficiency. Each cluster is handled independently. When buildings are merged in one cluster, data sets in other clusters can be ignored. Thus, the speed is accelerated, and cluster generation is a key step for our simplifying algorithm.

Many researches focused on building clustering for generalization. For example, Anders and Sester (2000) proposed a parameter-free cluster detection method. Li *et al.* (2004) made global and local building grouping before generalization, to implement the generalized models consistent with urban morphology and Gestalt theory. Chang *et al.* (2008) adopted a single-link hierarchical clustering to keep urban legibility. One limitation is that they just used Euclidean distance between the building footprints as the basis of clustering, which means to abstract the buildings into points without directions and shapes. Considering Euclidean distance as the only clustering criterion can guarantee that the clusters are localized. Obviously, it cannot describe the spatial relations among buildings accurately. We put forward a new kind of measure criterion, called the adjacent distance d_{adj} , as the distance metric for the single-link clustering algorithm, which is used to group building footprints into clusters.

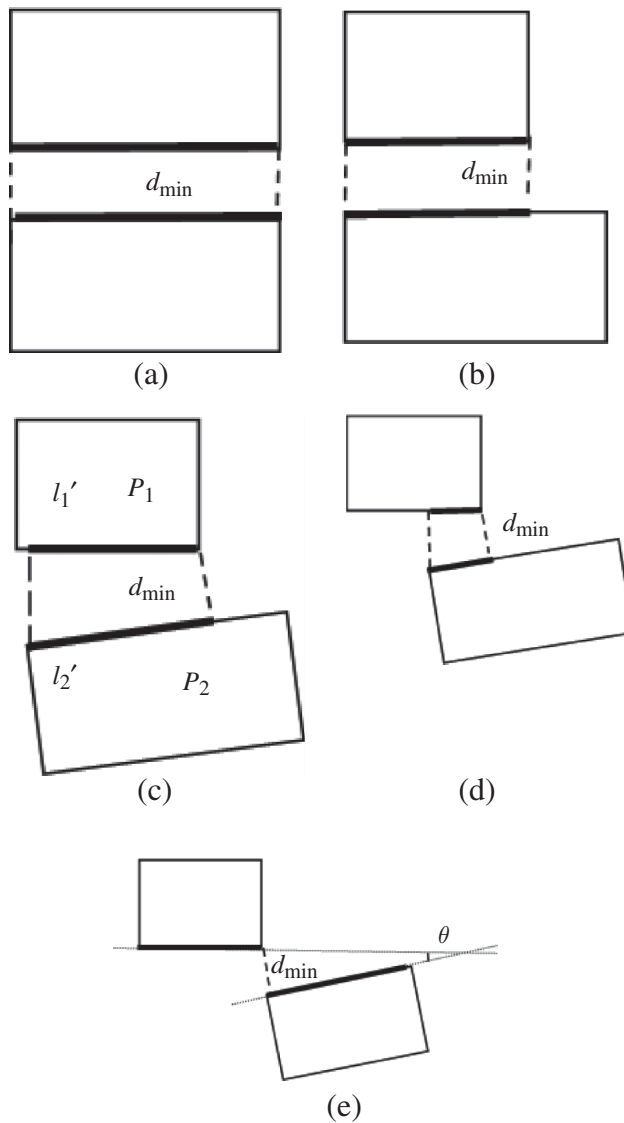


Figure 1. The adjacent sides (marked with bold lines) of polygons. d_{\min} is the distance between two polygons. From (a) to (e), d_{\min} remains unchanged.

3.1.1. Definitions

For the convenience of description, several definitions are stated as follows.

3.1.1.1. *The adjacent sides of two polygons.* If two polygons do not include or intersect with each other, their adjacent sides are calculated as follows:

- (1) Calculate the minimum distance d_{\min} between two polygons.

- (2) If d_{\min} is the distance between the vertex v_i of a polygon and the side $\overline{v_j v_{j+1}}$ (v_{j+1} is the next vertex of v_j , and v_{j-1} is the previous one) of another polygon, we calculate the angle formed by the sides $\{\overline{v_i v_{i+1}}, \overline{v_j v_{j+1}}\}$ and $\{\overline{v_i v_{i-1}}, \overline{v_j v_{j+1}}\}$ (the angle is denoted by θ in Figure 1e). Between $\overline{v_i v_{i+1}}$ and $\overline{v_i v_{i-1}}$, the one forming the smaller angle with $\overline{v_j v_{j+1}}$ is recorded as s_1 and with $\overline{v_j v_{j+1}}$ as s_2 ; s_1 and s_2 is a pair of adjacent sides.
- (3) If d_{\min} is the distance between the vertex v_i of a polygon and v_j of another polygon. The adjacent sides come from $\overline{v_i v_{i+1}}, \overline{v_i v_{i-1}}, \overline{v_j v_{j+1}}$ and $\overline{v_j v_{j-1}}$. We calculate the four angles formed by $\{\overline{v_i v_{i+1}}, \overline{v_j v_{j+1}}\}$, $\{\overline{v_i v_{i-1}}, \overline{v_j v_{j+1}}\}$, $\{\overline{v_i v_{i+1}}, \overline{v_j v_{j-1}}\}$ and $\{\overline{v_i v_{i-1}}, \overline{v_j v_{j-1}}\}$, respectively, and the pair of sides which forms the minimum angle is the adjacent sides.

3.1.1.2. *The adjacent distance between the polygons.* The lengths of the adjacent sides s_1 and s_2 are calculated and recorded as l_1 and l_2 . Project s_2 onto s_1 and get the projection length l'_1 . Similarly, project s_1 onto s_2 and get l'_2 . Then the adjacent distance d between the polygon p_1 and p_2 can be expressed as

$$d = d_{\min} \times (1 - N \times \lambda) \quad (1)$$

$$N = \frac{(l'_1/l_1 + l'_2/l_2)}{2} \quad (2)$$

where N represents the orientation and similarity relations of adjacent sides, λ is a weight factor, which ranges from 0 to 1. When λ is given and d_{\min} remains unchanged, d decreases with the increase of N . The pair of polygons which has the minimum value d should be clustered first. For two polygons, only if s_1 is parallel and completely opposite s_2 , $l'_1 = l_1$, $l'_2 = l_2$, $N = 1$, and correspondingly d has the smallest value. If adjacent sides do not parallel with each other, or have different lengths, or are staggered, d increases. Figure 1 gives an illustration of this process. Figure 1a has the minimum d , followed by Figure 1b–d. Figure 1e has the biggest d . From Figure 1, we see that adjacent distance reflects not only the distance between two buildings but also the relative position and directional relations. λ is used to determine the weight of N in d . The weight increases with increase in λ , which means that the relative position and directional relations should be taken into account. Empirically, we found that 0.6 was an appropriate value for λ .

3.1.2. Acquisition of adjacent sides

During calculation of the adjacent distance, acquiring adjacent sides is a key step. If the footprints p_i and p_j contain m and n vertices, respectively, the time complexity of acquiring adjacent sides is $O(mn)$. To improve the computing efficiency, we select parts of the vertices for calculating adjacent sides by means of two footprints' convex hull. The convex hull can be created by the Graham scan algorithm (Graham 1972), whose time complexity is $O(n \log n)$. Using the convex hull, the number of vertices participated in calculating the adjacent sides can be reduced as follows.

In most cases, a convex hull is composed of the vertices of two footprints (see Figure 2a). This convex hull can be divided into two types: the sides connecting vertices of the same footprints and the ones connecting the vertices of different footprints. The second kind of

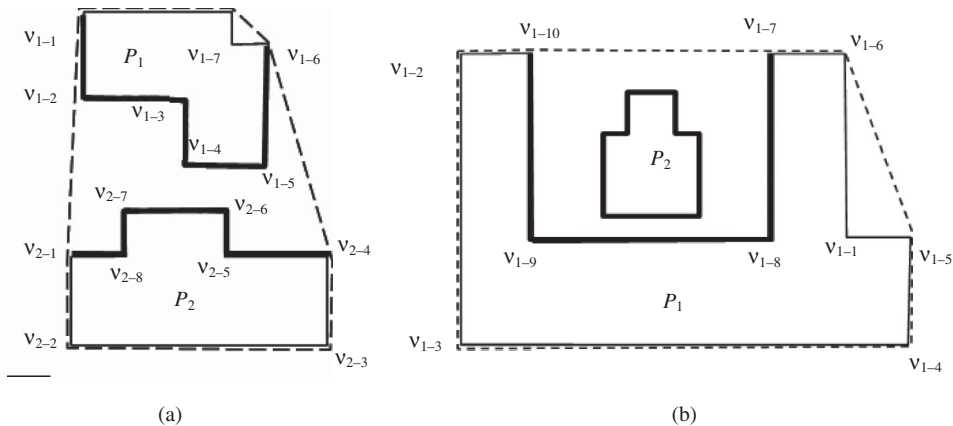


Figure 2. The convex hulls (dotted lines) of two polygons. (a) A convex hull composed of two footprints' vertices. (b) A convex hull composed of one footprint's vertices.

sides is called *boundary*. Each *boundary* has two endpoints. One is a vertex of p_i and another is a vertex of p_j . There are two *boundaries* in a convex hull, thus there are four endpoints correspondingly. These four endpoints divide the vertices p_i and p_j into two groups. One group can be ignored and the other should be preserved.

In Figure 2a, the sides $\overline{v_{1-1}v_{2-1}}$ and $\overline{v_{1-6}v_{2-4}}$ are two *boundaries*. v_{1-1} , and v_{1-6} divide vertices of p_1 into two groups. One group contains the vertices v_{1-1} , v_{1-2} , \dots , v_{1-5} and v_{1-6} , which are close to the footprint p_2 and should be involved in the calculation of adjacent sides. The other group contains the vertices v_{1-7} and v_{1-8} . These vertices are far away from p_2 and can be excluded from the calculation. Similarly, the vertices v_{2-2} and v_{2-3} of the footprint p_2 can be ignored. So, the vertices ultimately used to calculate the adjacent edges just include v_{1-1} , v_{1-2} , \dots , v_{1-5} and v_{1-6} of the footprint p_1 and v_{2-4} , v_{2-5} , \dots , v_{2-8} , v_{2-1} of the footprint p_2 . The calculation result shows that the side $\overline{v_{1-4}v_{1-5}}$ of p_1 and the side $\overline{v_{2-6}v_{2-7}}$ of p_2 are adjacent sides.

If the final convex hull is completely composed of one footprint's vertices, it means that the bigger footprint's convex hull contains the smaller footprint, as shown in Figure 2b. The bigger footprint and its convex hull form several gaps. The smaller footprint is certainly contained by one of these gaps. The vertices of this gap and the smaller footprint are used to compute the adjacent sides. In Figure 2b, p_1 and its hull form two gaps. One is composed of v_{1-7} , v_{1-8} , v_{1-9} and v_{1-10} , and the other is composed of v_{1-5} , v_{1-1} and v_{1-6} . p_2 locates in the first gap. Therefore, only the vertices of p_2 and v_{1-7} , v_{1-8} , v_{1-9} and v_{1-10} participate in calculating the adjacent sides.

3.2. Mergence and generalization of clusters

After the clustering, the footprints within every cluster need to be merged. Our process can be divided into two steps. The first one is to merge two footprints into a new footprint and the second one is to generalize the new footprint.

In the merging procedure, our aim is to minimize the empty spaces contained by the generated footprint. This process inevitably introduces geometric errors, because the new generated footprint contains previous empty spaces. The points that participated in calculating adjacent sides are reused here to build the constrained Delaunay triangles (in Figure 3a,

the dotted lines illustrate the generated triangles). We define a triangle pair (T_1, T_2) called a connection pair. For the polygons p_1 and p_2 , (T_1, T_2) is a connection pair, if

- (1) T_1 and T_2 share a side, and this side connects p_1 and p_2 ;
- (2) another side s_1 of T_1 connects two vertices of p_1 and
- (3) another side s_2 of T_2 connects two vertices of p_2 .

For each connection pair, the adjacent distance d between s_1 and s_2 are marked as d_{pair} . According to the definition, the triangle of a connection pair has a side that connects two vertices of the same footprint. There are two cases for this side. One case is that this side connects two adjacent vertices. The other case is that the two vertices are not adjacent in the footprint. In Figure 3a, the triangles $\{v_{2-4}, v_{1-5}, v_{1-6}\}$ and $\{v_{2-4}, v_{2-6}, v_{1-5}\}$ form a connection pair. $\overline{v_{1-5}v_{1-6}}$ is a side of p_1 and satisfies the first case. The side $\overline{v_{2-4}v_{2-6}}$ is the second case. If two triangles of a connection pair both meet the first case then this pair is called a strict connection pair. Otherwise, the pair is called a non-strict connection pair. If two footprints p_1 and p_2 have one or more strict connection pairs, the one with the minimum adjacent distance d_{pair} is used to connect the two footprints. For example, in Figure 3a, the pair $\{v_{1-4}, v_{1-5}, v_{2-6}\}$ and $\{v_{1-4}, v_{2-6}, v_{2-7}\}$ has the minimum adjacent distance. The new generated footprint is illustrated in Figure 3c. If two polygons have no strict connection pairs, we check non-strict connection pairs and the one with the minimum d_{pair} is used to connect the two footprints. The graph in Figure 3b has no strict connection pairs, and the non-strict

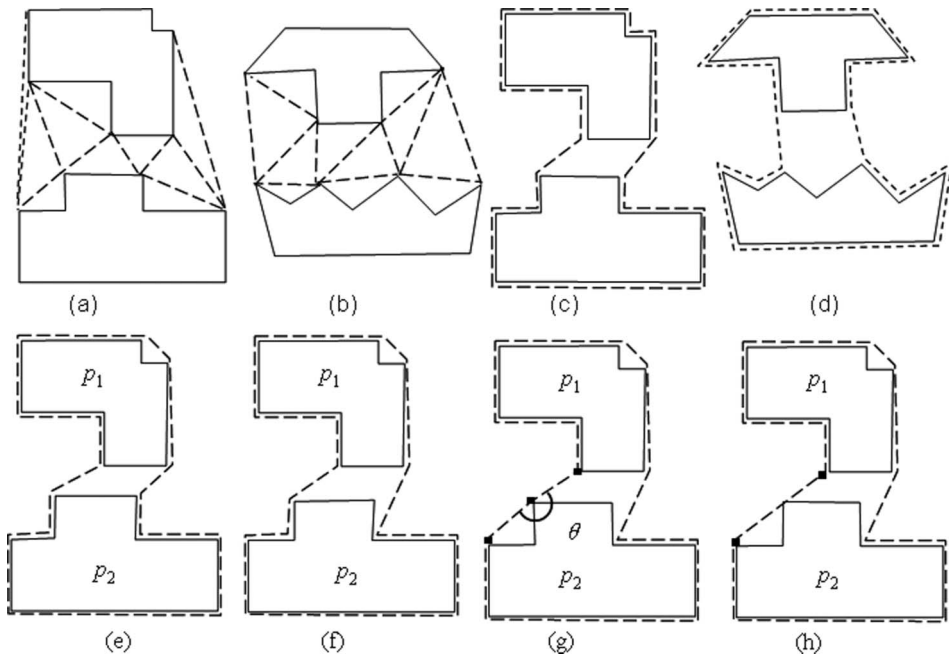


Figure 3. The merging process of two building footprints. (a), (b) Two original footprints and Delaunay triangles. (c), (d) Merging results. (e)–(h) Four generalization steps for (c).

connection pair $\{v_{1-4}, v_{1-5}, v_{2-8}\}$ and $\{v_{1-5}, v_{2-4}, v_{2-8}\}$ has the minimum d_{pair} . The new generated footprint is illustrated in Figure 3d.

The new footprint may have too many geometric details and can be further generalized. We employ the polyline generalization algorithm proposed by Visvalingam and Whyatt (1993) to remove small concave triangles. Assume v_{i-1} , v_i and v_{i+1} are consecutive vertices of a footprint in anti-clockwise direction. They compose a triangle $\{v_{i-1}, v_i, v_{i+1}\}$. If the direction starts from v_{i-1} to v_i , then to v_{i+1} , and finally returns to v_{i-1} clockwise, this graph is a concave triangle, and v_i is a middle vertex. Each vertex with its neighbouring two vertices forms a triangle. If it is a concave triangle, its area is recorded. After all vertices of the new footprint are traversed, we obtain the concave triangle with the minimum area. If the area is smaller than a threshold, the middle vertex of this triangle is removed from the footprint. Then, the process of area calculation and vertex deletion is repeated until there is no concave triangle, or every concave triangle's area is larger than the threshold. Figure 3e–g illustrates the process. The threshold is calculated by the following formula

$$\delta a = \frac{a_{\text{ave}}}{\lambda} \quad (4)$$

Hierarchical clustering process can be represented by a tree structure. Assume the new footprint's two children locate at the level l of the tree, and then a_{ave} is the average area of all the footprints at the level l . λ is a user-defined variable. We found empirically that a value of 50 for λ is sufficient. The threshold is calculated for each cluster so that it fits the perceptual nature of the hierarchy during the LOD establishing process.

After small concave triangles are removed, the new generated footprint's vertices are traversed again to further reduce the vertices. This time, we check every two consecutive sides. If the angle (denoted by θ in Figure 3g) formed by them is larger than a threshold (empirically, we use 175°), the shared vertex by the two sides is removed. This process is shown in Figure 3g and h.

As we can see from Figure 3, our method has achieved a good balance in reducing details and maintaining the profiles of the original footprints. It is also consistent with visual psychology, that is, when the neighbouring elements are merged into a whole, the profile curve is close to the *boundaries* of the targets. At the same time, this approach avoids too many empty spaces being merged into the final footprint. Therefore, the distinguishable objects used for the orientation, such as the plaza, are preserved during the generalization. Without introducing new vertices, the above merging process saves much memory.

The footprints of 2.5D building models on the xy plane and the height can be handled separately during the merging process. For the new created cluster merged by n buildings, its height is computed by Equation (5):

$$h = \frac{\sum_{i=1}^n (h_i \cdot a_i)}{\sum_{i=1}^n a_i} \quad (5)$$

where h is the height of this cluster, h_i the height of the building i and a_i the weight coefficient, representing the footprint area of the building i .

Compared to the methods detecting the topological relations based on Delaunay triangulation (Li *et al.* 2004, Royan *et al.* 2006), our approach ignores the topological relations during the clustering. In addition, we only consider the involved two footprints and ignore their relations with other buildings during the merging process. Therefore, a new generated footprint may intersect with or be contained in other footprints. The intersecting footprints

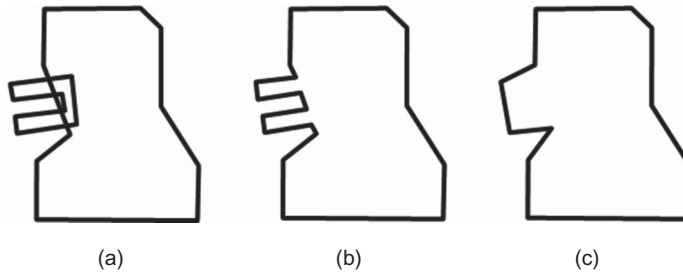


Figure 4. The merging process of two intersecting footprints. (a) Two original footprints. (b) Merging result. (c) Generalization result.

can be merged similarly with the above procedure. The only difference is that the intersection points are first computed and inserted into the new footprint as shown in Figure 4a. Then the new footprint is generalized using the same method. The height is calculated from Equation (5). For the inclusion cases, we just take the larger footprint as the new cluster, whose height can be computed also by Equation (5). Although our method introduces some topological error occasionally, experimental results show that this type of intersection and inclusion has no obvious effects on urban building visualization.

3.3. Establishment of LOD models

3.3.1. Data storage

We use a tree structure to store the simplified models. Because there are no new vertices introduced, the coordinates of the original vertices are stored alone. For every node in the tree, we only store the identifiers of its corresponding vertices rather than its vertices' coordinates (see Figure 5). It can avoid vertices' coordinates being stored redundantly, thus saving much storage space. Every node also stores the attached useful information, such as the sum of the building area and its height.

3.3.2. Generation of the hierarchical tree

The single-link clustering ensures the clusters obey paths and edge. However, only the two nearest nodes are merged every time. Therefore, the generated tree has too many levels. For an urban with n buildings, the generated tree has n levels. For n building models, the ideal depth is $\log(n)$ (Chang *et al.* 2008). Thus, we want to reduce the tree's levels to n' , and n' can be set as $\log(n)$, or $\log(n)/\lambda$, where λ is a user-defined variable and $\lambda > 1$. By giving $n'-1$ distance thresholds, the tree depth can be reduced to n' . These thresholds may be designated by artificial interference. But it is more reasonable to compute them automatically according to the features of building models.

In the real world, for adjacent buildings that locate at the same district, the distances between them are small, whereas the distances between different districts may be much larger. When clustering, if the distance of the two nodes d is dramatically larger than that of the previous merging step, it usually means that this step is to merge two clusters representing different districts, whereas the previous step is to merge buildings in the same district. Therefore, it is appropriate to distinguish this step from previous ones. We create a threshold

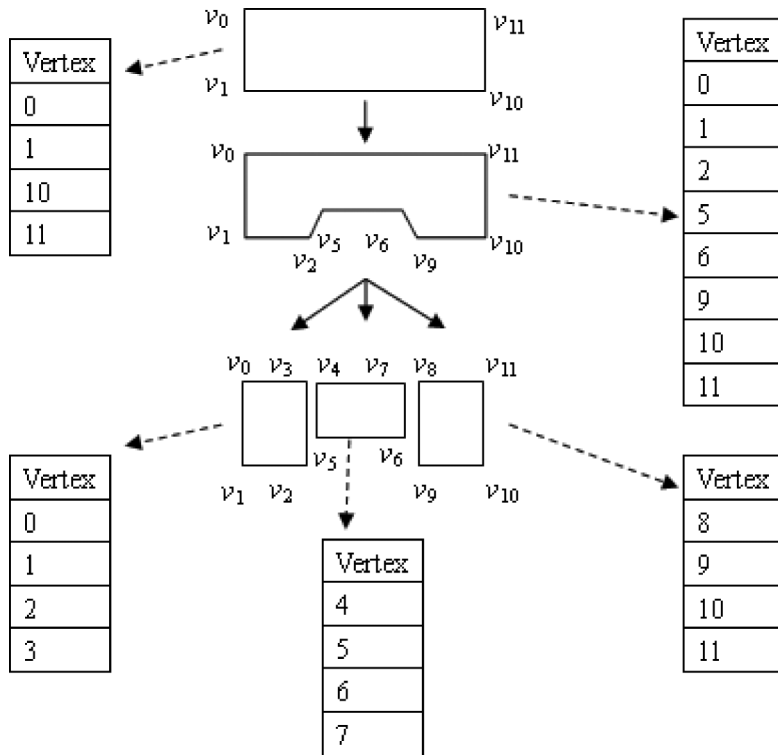


Figure 5. The hierarchical models are stored in the tree structure. The identifier of each node is stored in the vertex array.

that is smaller than this step's d and larger than the previous step's d . Thus, the nodes generated by the previous steps can be merged into one node.

In Figure 6a, building footprints are clustered by the order shown in Figure 6b. If we want to reduce the tree's levels to three, Figure 6c and d are two possible results. Figure 6c appears better than Figure 6d. Figure 6b also shows each step's d , they are 19, 23, 27, 32 and 58, respectively. d increases smoothly at the beginning but has a rapid rise from 32 to 58. Figure 6c is obtained by setting a distance threshold that is larger than 32 and smaller than 58.

Therefore, we give a threshold when a step's d is dramatically larger than that of the previous merging step. During the single-link clustering, we record each merging step's adjacent distance d . After clustering, these distances are sorted in ascending order to generate a *distance sequence*, and we compute the distance increasing rate r using this *distance sequence* (see Figure 7):

$$r_i = \frac{(d_{i+1} - d_i)}{d_i} \tag{6}$$

where d_i and d_{i+1} stand for two successive values in *distance sequence*.

We select the largest $n'-1$ r to generate $n'-1$ thresholds. For the rate r_i , the threshold is set as $(d_i + d_{i+1})/2$, or any value between d_i and d_{i+1} . The $n'-1$ distance thresholds are sorted in ascending order to generate a *threshold sequence* (see Figure 7). By this way, the tree's depth is reduced to n' . However, the children number of each non-leaf node may be greatly

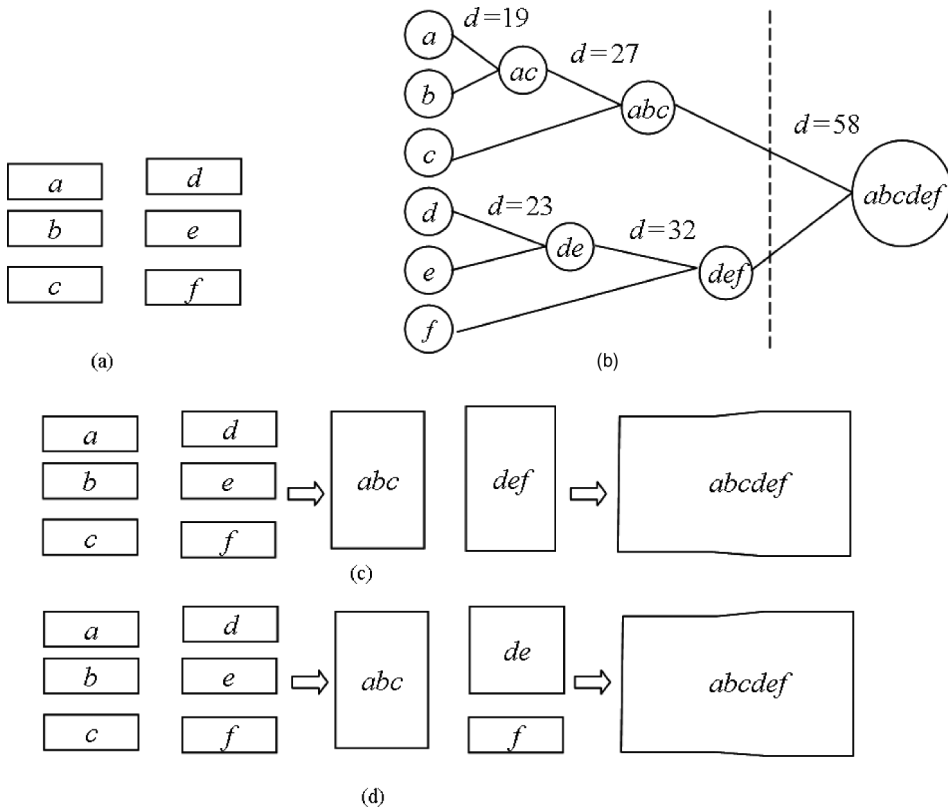


Figure 6. (a) Original fingerprints. (b) Clustering order. d is the distance used in each clustering step. (c) and (d) are two possible results, if the tree's depth is reduced to 3.

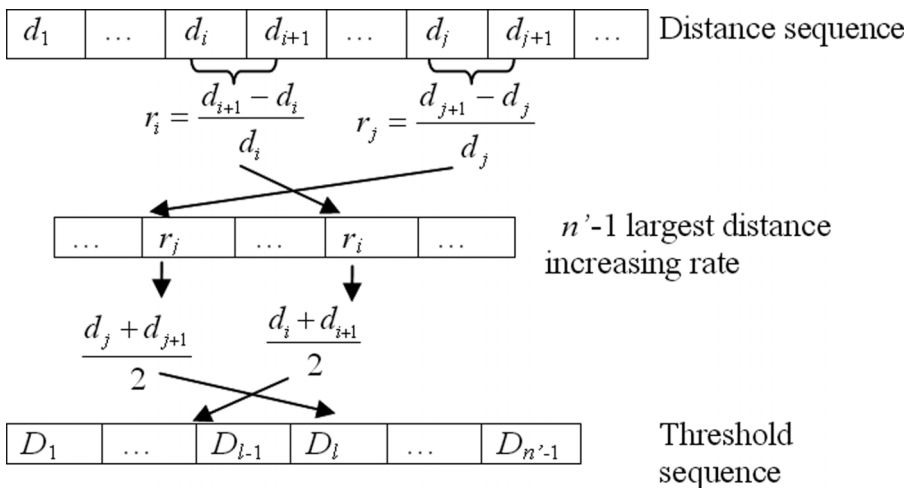


Figure 7. Flow chart for generating thresholds.

different. Therefore, the tree needs to be further balanced. For a completely balanced tree with n' levels, each non-leaf node has p children. p is computed from Equation (7):

$$p = \text{int}({}^{n'}\sqrt{n}) \quad (7)$$

The $\text{int}()$ function rounds up a number to an integer value.

To prevent a node having too many children, we set an upper bound δp for a node's children number. δp can be set slightly larger than p , such as $2p$. Thus, if a level has m nodes and these nodes have m_{ch} children, m_{ch} should not be larger than $m \times \delta p$. The thresholds in the *threshold sequence* are used to condense levels in order from the lowest value to highest one. If the new generated level does not meet the condition $m_{\text{ch}} \leq m \times \delta p$, it means that the span between D_l (the threshold D_l is used to condense several levels to a new level) and the prior threshold D_{l-1} is too large. Thus, we insert a new threshold D_{new} between D_{l-1} and D_l . D_{new} is generated as follows: First, we extract a sub-array that falls in the interval $[D_{l-1}, D_l]$ from the *distance sequence*. Second, the maximum distance increasing rate r_{max} in the sub-array is used to generate D_{new} . Finally, we use D_{new} to generate a new level and check if this level meets $m_{\text{ch}} \leq m \times \delta p$. If not, we use the interval $[D_{l-1}, D_{\text{new}}]$ to regenerate a new threshold to replace D_{new} . Repeat this process until the level generated by the new threshold meets $m_{\text{ch}} \leq m \times \delta p$.

4. Visualization of large-scale urban models

We have established LOD urban building models. These building models near the viewpoint have a high level of detail, whereas the far ones can be rendered by coarse models. We use the amount of pixel errors as the criterion of LOD division similarly as Chang *et al.* (2008) did. Pixel errors are the differences between original buildings and coarse models on computer screens. The pixel errors should be controlled in a certain value.

In the visualization process, the source of errors that are obviously noticeable can be divided into two main categories. One is related to the height. Although the children nodes have different heights, their father node has only a height value. The other is related with the area. The vacant regions among the buildings are included in the simplified polygons. So, the area of a father node's roofs is larger than the sum of original buildings' roofs.

For a non-leaf node, define the height difference $\Delta h = h_{\text{max}} - h_{\text{min}}$. h_{max} and h_{min} are, respectively, the heights of the tallest and the lowest original buildings contained by the node. During runtime, we project a user-defined height tolerance δh_{screen} onto each cluster mesh and convert it to a height value in the world coordinates (called δh). Similarly, the area difference Δa is defined as the subtraction of the area between a node's footprint and the sum of its original buildings' footprints. A user-defined height tolerance δa_{screen} is projected onto each cluster mesh's roof to get δa in the world coordinates. During the navigation, for a non-leaf node, if $\Delta h > \delta h$ or $\Delta a > \delta a$, it is not rendered, and its descendants are checked recursively.

This should be different from the methods that used the distance to establish the LOD criterion. This approach not only takes into account the distance between the viewpoint and building models but also considers the geometrical error of the cluster itself. Large physical objects often act as landmarks in the environment, and such landmarks are usually much taller than their surrounding objects. For the node merged by a landmark and the surrounding buildings, its Δh is always large. Thus, landmarks will be rendered even it is far away from the viewpoint.

5. Experiments

To demonstrate the performance and efficiency of our methodology, experiments have been done for purposes of evaluation. The experiments are performed on a personal computer with 3.00 GHz Intel (R) Pentium (R) 4 CPU, 768 MB main memory and ATI mobility Radeon X300 graphic card. Indeed, the evaluation of visualization results is a very difficult issue. The adequacy of a generalization and visualization result is dependent on various factors such as the algorithms and constraints used to select the algorithms and operations (Li *et al.* 2004), and it is difficult to quantify a person's sense of spatial awareness in an urban environment (Chang *et al.* 2008).

The experiment results obtained by adopting adjacent distance and traditional distance are first presented. Figure 8a is a graphic illustration of the original building footprints. It has 37 buildings, and we hope its depth is $\text{int}(\log(37)/1.5)$. Figure 8b–d shows the results when the adjacent distance d_{adj} is taken as the clustering factor. Figure 8e–g illustrates the results just using the traditional distance between polygons d_{mini} . The automatically generated distance thresholds are shown in Table 1. As shown in Figure 8, the adjacent distance can better take into account the common orientation continuity and similarity among the buildings, as well as better preserve road information.

The comparison between the time complexity of our method and that of the one proposed by Chang *et al.* (2008) is also made. The time complex of their method is $O(n^3)$ in the worst case. In our method, Delaunay triangulation algorithm has a worst-case upper bound $O(n^2)$.

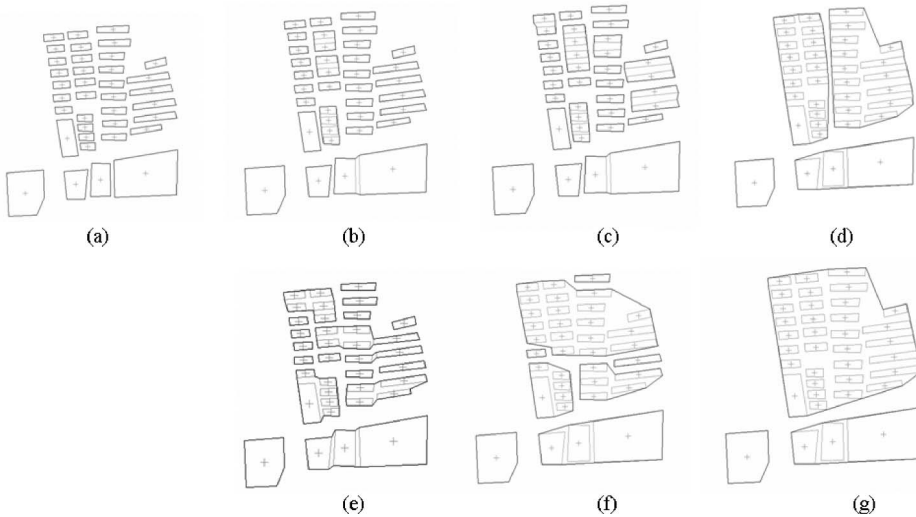
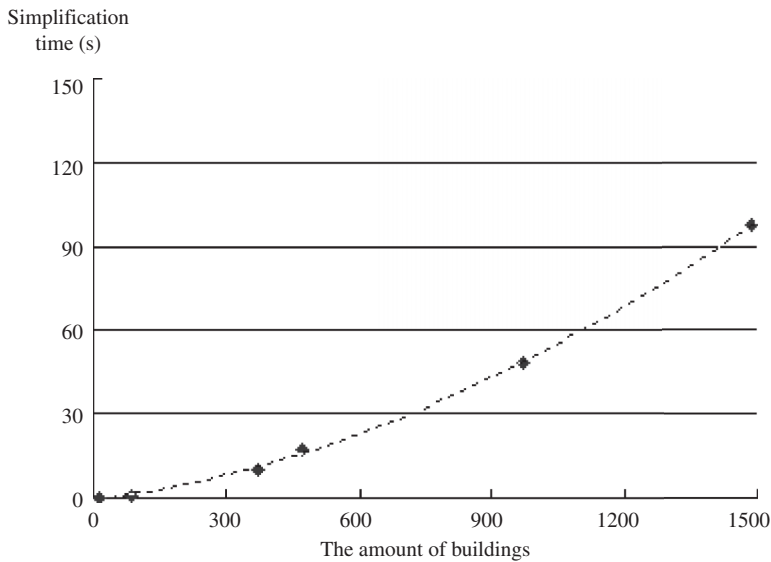


Figure 8. Examples of footprint simplification. (a) The original building footprints. (b)–(d) show the clustering results using the traditional distance. (e)–(g) illustrate the results using the adjacent distance.

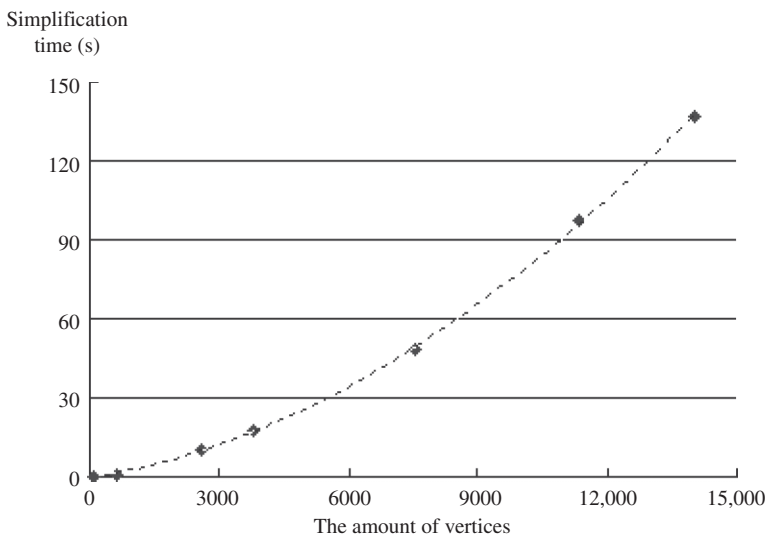
Table 1. The distance thresholds and number of nodes in Figure 8.

	a	b	c	D	e	f	g
Distance threshold	–	1.2	2.0	2.9	8.6	10.2	25.1
Number of nodes	37	31	26	4	18	8	3

Our approach reduces the time complexity of the algorithm to $O(m^2)$, where m is the number of vertices in Delaunay triangulation. The time complexity for generating convex hulls is $O(n \log n)$. The time complexity for getting the connection pair is also $O(m^2)$. The time complexity of Visvalingam and Whyatt's polyline generalization algorithm (1993) is $O(n \log n)$. So in the merging process, overall time complexity of our method is less than $O(n^2)$. Figure 8 illustrates the relations between the generalization time and amount of buildings, as well as vertices of 2D footprints. The dashed line in Figure 9 is the quadratic



(a)



(b)

Figure 9. (a) Simplification time and amount of buildings. (b) Simplification time and amount of vertices.

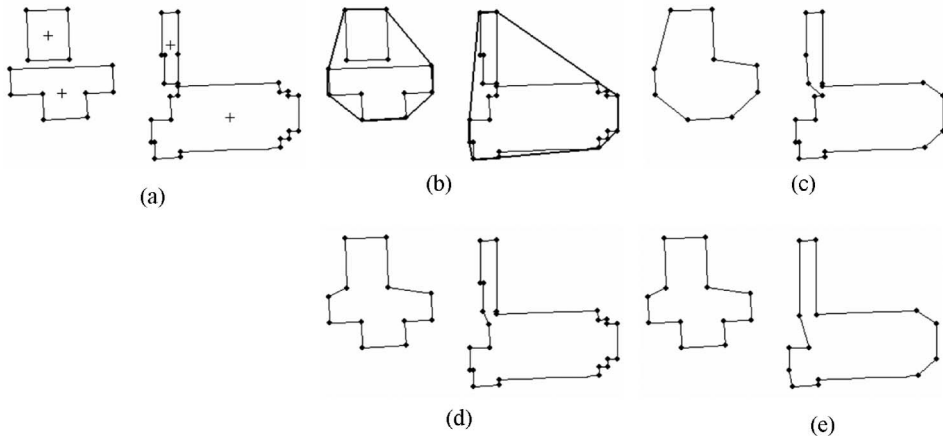


Figure 10. Examples for footprints generated by the proposed algorithm and Chang *et al.* (2008) algorithm. (a) Original building footprints. (b) Convex hulls of clusters. (c) New generated footprints using Chang *et al.* (2008) algorithm. (d) New footprints generated by the proposed algorithm. (e) Generalized footprints by the proposed algorithm.

curve fitting by the experimental data. It can be seen that the simplification time grows at quadratic rate of data set volume growth.

Figure 10 compares the merging results generated by the proposed method against Chang *et al.* (2008) method. Chang *et al.* (2008) first found out the convex hull of the two footprints in a cluster (see Figure 10b), then iteratively subdivided the convex hulls' line segments and stopped the subdivision process when the parent cluster's number of sides reaches 75% of the two children's sides. However, the generalization degrees may be very different with different clusters. Because the children's shape and side number vary greatly for different clusters, when the subdivision process stopped, in some clusters, many geometric details have been removed; in some other clusters, most details may still remain. For example, in Figure 10c, we can see that more geometric details remain in the right cluster, compared with the left cluster. In contrast, our method first generates footprints by connecting two children's footprints (see Figure 10d), then the footprints at the same level of the hierarchical tree are generalized using the same area threshold δ_a . Thus, our method can get similar generalization degrees for different clusters (see Figure 10e).

To further prove the feasibility of our method, take the building data sets of Chaoyang District, Beijing, China, as a test. The data set contains 35,138 buildings and 513,603 footprint vertices. We set the hierarchical tree's depth as $\text{int}(\log(35,138)/1.5)$, which is approximately 9. The final generalization results are stored in a hierarchical tree, and the tree has 12 layers. It takes 1 hour and 2 minutes to finish the generalization.

During the rendering process, we set the height difference threshold δ_H as 20 pixels, and the threshold of new adding roof areas δ_A as 200 pixels. Figure 11 is the views of the scene. Figure 11a, c and e shows the rendering results of the original building models. Figure 11b, d and f shows the corresponding LOD models rendered using our hierarchical method. Figure 12 illustrates the curves of frame rates for rendering the building models with/without our hierarchical method. From these two figures, we can see our proposed approach effectively reduces the rendering time and makes a good balance between rendering efficiency and visual quality. Meanwhile, it can maintain the districts, roads, landmarks

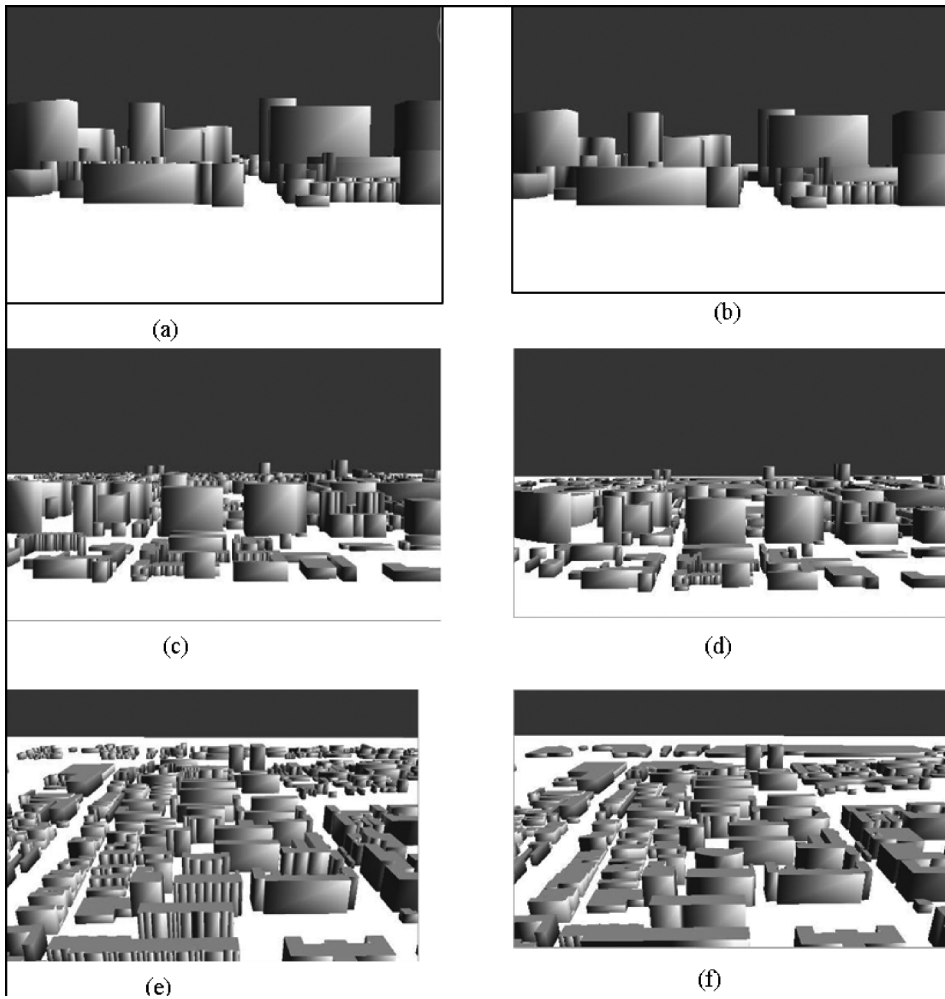


Figure 11. Three-dimensional building models rendering. (a), (c) and (e) are the original building models. (b), (d) and (f) are the corresponding LOD building models rendered by our approach.



Figure 12. Comparison of frame rates between rendering the origin building models and LOD models.

Downloaded by [Institute of Remote Sensing Application] at 23:29 08 November 2011

and other elements of urban legibility. The topological errors among the clusters influence little on the final visual effects.

6. Conclusion

This article mainly focuses on generalizing and visualizing urban building models in the context of Gestalt psychology and urban legibility. The article puts forward a new distance measurement method for single-link clustering and introduces new approaches for merging clusters and automatically generating the distance thresholds aiming at building up the hierarchical tree. Finally, a criterion for selecting appropriate levels of details in interactive urban visualization is presented. The experiments show that the clustering result follows the Gestalt organization laws, including common orientation, continuity and similarity to a certain degree, and the important features of urban legibility, such as roads and landmarks, are well preserved at the rendering process. Compared with the algorithm which extracts the axes of the roads and rivers first (Li *et al.* 2004), our proposed approach in this article is simple, easy and suitable for handling with massive building models.

The main contributions of our article are the following: (1) a hierarchical tree is presented to store the final footprints of the clustering generalization; (2) the adjacent distance is proposed as the clustering criteria and (3) a new way is proposed to merge clusters based on Delaunay triangulation and the polyline generalization algorithm.

7. Discussion and outlook

In this article, the clustering process ignores the topological relations, resulting in topological errors occasionally. It is not allowed in 2D map generalization or some other areas. However, we aim to maintain the urban legibility during 3D urban building visualization. The intersections and inclusions among the clusters do not cause obvious aliasing. The benefit for only considering the distance in the footprints is that the clusters are guaranteed to be localized. When additional information is included, the clustering could become dislocated.

Our approach still has potential for improvements and many parameters are chosen intuitively instead of being defined by users. Many directions open up for further research. On the one hand, the future study is to consider how to include additional considerations. On the other hand, how to get simplified city models which are coincident with human special cognition still remains an open problem. Because it is difficult to quantify a person's sense of spatial awareness, more research work needs to be done for generating a humanized 3D virtual urban system.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 60736007 and 60972128) and 973 Program (No. 2007CB714403). The authors would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions.

References

- Allouche, M. and Moulin, B., 2005. Amalgamation in cartographic generalization using Kohonen's feature nets. *International Journal of Geographical Information Science*, 19 (8), 899–914.

- Anders, K.H., 2005. Level of detail generation of 3D building groups by aggregation and typification. *In: Proceedings 22nd international cartographic conference*, 9–16 August 2005, La Coruña, Spain, ICA, Canadian Institute & Geomatics, 9–16.
- Anders, K.H. and Sester, M., 2000. Parameter-free cluster detection in spatial databases and its application to typification. *ISPRS Archives for Photogrammetry and Remote Sensing*, 23, Part 4 [CD-ROM].
- Bai, F.W. and Chen, X.Y., 2001. Generalization for 3D GIS. *In: J. Chen, et al., eds. Proceedings of the 3rd ISPRS workshop on dynamic and multi-dimensional GIS*, 23–25 May 2001, Bangkok, Thailand, ISPRS Press, 8–11.
- Brassel, K. and Weibel, R., 1988. A review and framework of automated map generalization. *International Journal of Geographical Information Systems*, 2 (3), 229–244.
- Chang, R., et al., 2008. Legible simplification of textured urban models. *IEEE Computer Graphics and Applications*, 28 (3), 27–36.
- Cignoni, P., et al., 2007. Ray-casted BlockMaps for large urban models visualization. *Computer Graphics Forum*, 26 (3), 405–413.
- Fan, H.C., Meng, L.Q., and Jahnke, M., 2009. Generalization of 3D buildings modelled by CityGML. *In: Advances in GIScience, proceedings of the 12th AGILE conference*, 2–5 June 2009, Hannover, Germany, New York: Springer, 387–405.
- Forberg, A., 2007. Generalization of 3D building data based on a scale-space approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62 (2), 104–111.
- Forberg, A. and Mayer, H., 2002. Generalization of 3D building data based on scale-space. *International Archives of Photogrammetry and Remote Sensing*, XXXIV (4), 225–230.
- Glander, T. and Döllner, J., 2008. Automated cell based generalization of virtual 3D city models with dynamic landmark highlighting. *In: Proceedings of the 11th ICA workshop on generalization and multiple representation*, 20–21 June 2008, Montpellier, France, Canadian Institute & Geomatics.
- Glander, T., and Döllner, J., 2009. Abstract representations for interactive visualization of virtual 3D city models. *Computers, Environment and Urban Systems*, 33 (5), 375–387.
- Graham, R.L., 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 26, 132–133.
- Gröger, G., Kolbe, T.H., and Czerwinski, A., 2006. Candidate OpenGIS CityGML implementation specification. Technical Report, Open Geospatial Consortium, Inc.
- Harrie, L., 1999. The constraint method for solving spatial conflicts in cartographic generalization. *Cartography and Geographic Information Science*, 26 (1), 55–69.
- Kada, M. and Luo, F., 2006. Generalisation of building ground plans using half-spaces. *In: Proceedings of the international symposium on geospatial databases for sustainable development*, 25–30 September 2006, Goa, India. ISPRS Technical Commission IV.
- Li, Z., et al., 2004. Automated building generalization based on urban morphology and Gestalt theory. *International Journal of Geographical Information Science*, 18 (5), 513–534.
- Li, Z.L., 2007. *Algorithmic foundation of multi-scale spatial representation*. Bacon Raton, FL: CRC Press (Taylor & Francis Group), pp. 281.
- Lynch, K., 1960. *The image of the city*. Cambridge, MA and London, UK: MIT Press, Massachusetts Institute of Technology, 5–8.
- Meng, L.Q. and Forberg, A., 2007. 3D building generalisation. *In: W. Mackaness, A. Ruas and T. Sarjakoski, eds. Challenges in the portrayal of geographic information*. Amsterdam, The Netherlands: Elsevier Science.
- Regnauld, N., 1996. Recognition of building clusters for generalization. *In: M. Molenaar and M.J. Kraak, eds. Advances in GIS II*. London: Taylor & Francis, 185–198.
- Regnauld, N., 2001. Contextual building typification in automated map generalization. *Algorithmica*, 30, 312–333.
- Royan, J., Balter, R., and Bouville, C., 2006. Hierarchical representation of virtual cities for progressive transmission over networks. *In: 3DPTV'06: proceedings of the third international symposium on 3D data processing, visualization, and transmission*, 14–16 June 2006, University of North Carolina, Chapel Hill, USA, IEEE Computer Society Press, 432–439.
- Royan, J., Bouville, C., and Gioia, P., 2003. PBTree: a new progressive and hierarchical representation for network-based navigation in urban environments. *In: VMV'03: proceedings of the vision, modeling, and visualization conference (VMV 2003)*, 19–21 November 2003, München, Germany, Aka GmbH, Germany, 299–307.

- Sester, M., 2000. Generalization based on least squares adjustment. *International Archives of Photogrammetry and Remote Sensing*, 33, 931–938.
- Thiemann, F., 2002. Generalization of 3D building data. In: *Proceedings of ISPRS commission II symposium 'geospatial theory, processing and applications'*, Ottawa, Canada. *International Archives of Photogrammetry and Remote Sensing*, 34(Part 4) [CD-Rom].
- Visvalingam, M. and Whyatt, J.D., 1993. Line generalisation by repeated elimination of points. *The Cartographic Journal*, 30 (1), 46–51.
- Wertheimer, M., 1923. Law of organization in perceptual forms. In: W.D. Ellis, ed. *A source book of Gestalt psychology*. London: Kegan Paul, Trench, Trubner, 71–88.