



Automatic simplification and visualization of 3D urban building models

Jinghan Xie^a, Liqiang Zhang^{a,*}, Jonathan Li^b, Hao Wang^a, Ling Yang^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 100875, China

^b Department of Geography & Environmental Management, Faculty of Environment, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada

ARTICLE INFO

Article history:

Received 11 June 2011

Accepted 9 January 2012

Keyword:

Urban buildings

Generalization

Urban legibility

Dynamic visualization

ABSTRACT

A high-fidelity and real-time rendering urban building model was implemented through the simplification of 3D building groups. An approach to simplify a single 3D building has been addressed for generating a levels-of-detail (LOD) building model in nearby urban regions. In farther regions, a single-chain cluster was used to collect footprints of neighboring building groups. To effectively merge footprints, the Delaunay triangulation and line simplification were employed. As a result, a coarse LOD model was created based on generalized footprints and building heights. Our approach not only preserved urban legibility (Lynch, 1960), which is effective for viewers visually navigating through an urban environment, but also implemented dynamic visualization of 3D city models.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

3D urban visualization has been seen as a fast-growing research topic. The development in remote sensing and modeling geospatial data has helped to enhance visual clarity of urban models. Compared with 2D, 3D visualization helps to create a more realistic environment where users can develop a better sense of spatial awareness. For instance, in urban planning, analyzing in a 3D environment is more efficient than analyzing in 2D (Pu and Vosselman, 2009). To conveniently view urban models in real time on any computer, more information and better accessibility to city models would be required. Real-time visualization of cities is a challenging task because a city is comprised of numerous of buildings. The simplification and reduction of city objects is important for applications in mobile devices, such as calculating visibility graphs in navigation applications (Mao et al., 2011). Generalization operations are usually embedded in 3D data models, especially for creation of models represented by levels-of-detail (LOD). Furthermore, they can also be performed in real time. During generalization and visualization of large building models, the geometric relationship between models and human spatial cognition is important and should be considered. On the premise of keeping spatial geometrical precision, render of urban building models, which fits people's vision habit and urban legibility, can help people gain a comprehensive view of the structures. Therefore, it becomes necessary to apply the generalization principles in order to effectively communicate spatial information.

In this paper, a combination of single 3D building simplification and abstraction of building groups is addressed to provide real-time rendering of large-scale 3D urban building models. Our approach not only preserves urban legibility (Lynch, 1960), which is effective for viewers visually navigating an urban environment, but also implements dynamic visualization of 3D city models.

The remainder of our paper is structured accordingly: After reviewing past studies in Section 2, we will describe the approach for simplification and aggregation of city building models including simplification of single buildings and generalization of building groups in Section 3. Experimental results can be found in Section 4. A summary of the paper including future recommendations will be discussed in Section 5.

2. Related work

Meng and Forberg (2007) presented an overview of 3D generalization issues. They provided a basis from which 2D operations, such as aggregation, typicality and landmark exaggeration (Bai and Chen, 2001), can be extended to handle with 3D generalization problems. Anders (2005) proposed an algorithm for simplifying 3D urban models by aggregating nearby building models. He projected building models onto three orthogonal planes and obtained simplified models based on the projection. However, the algorithm is only suitable for simple symmetric models that do not self-occlude during the projection. A scale-space approach was introduced to generate LOD representations of 3D city models (Forberg and Mayer, 2002; Forberg, 2007). This approach is employed for orthogonal building structures; it works by moving parallel facets toward each other until the facets are merged. Thiemann (2002) proposed to decompose a building into basic 3D primitives. These primitives

* Corresponding author.

E-mail address: zhanglq@bnu.edu.cn (L. Zhang).

are organized according to they are constructive solid geometry (CSG) representations which are subsequently generalized on the CSG-tree. Thiemann and Sester (2006) described adaptive 3D templates. They classified urban buildings into a limited number of types with characteristic shapes. A generic set of templates were employed to replace the original 3D shapes with the most similar of those templates. Since the semantics of the template are known, specific features of the buildings can be emphasized efficiently. Kada (2005) presented an approach for generalizing 3D building models. His approach remodels a building model with only a few planes optimized toward the original building. This simplifies roof geometry while preserving its coplanar, parallel and orthogonal features. In the Kada (2007)'s approach, a building complex was first divided into cells using main lines of its footprint. The borders of these cells were then used to form facades to simplify the model. To simplify the structure of building roofs, feature detection was applied explicitly by instancing roof shape primitives for each cell and selecting the cell with the best-fit. In addition, Kada (2011) extended the building generalization and aggregation approach based on the cell decomposition to use morphological operations on a raster representation of the initially vectorial data. This approach effectively classified cells into categories (i.e. building and non-building cells), simplified buildings and organized into building groups. It was found that repeated execution of morphological operations may change the shape of an object's boundaries.

Aliaga et al. (2007) proposed a method where photographed and subdivided buildings are used to construct a representative grammar by automatically finding repetitive patterns of the building features. For the creation of city tourist maps, Grabler et al. (2008) applied several techniques for building simplification, optimization, displacement and labeling. This approach creates pleasing but static oblique images of a city. Royan et al. (2006) applied a Delaunay triangulation-based merging algorithm and a binary-tree structure called PBTtree to represent densely urban areas over the network. This allowed users to navigate the city freely. However, the many levels of the generated PBTtree inhibit the selection of efficient building models. Cabral et al. (2008) presented a coupled texture, geometric approach for modeling architectural scenes by reshaping and combining existing textured models. This approach is effective for analyzing small sets of the pre-existing model "pieces" but not large sets containing models of various characteristics. Chang et al. (2008) proposed a view-dependent visualization of urban buildings that maintains the legibility of the city. In this method, buildings are organized into clusters; this allows the extraction and simplification of the outer layer surrounding every building's footprint within the cluster. The simplified hull is then extruded to its contained weighted average building height. Glander and Döllner (2008) proposed an infrastructure network to determine landmark buildings in small scales. Their network is an interactive visualization that dynamically highlights landmark buildings. They used a single building block to replace individual models and dynamically exaggerated global landmark objects. Therefore, the borders of single buildings are invisible from the observers, while preserving local landmarks in their representation, if needed. However, most visible landmarks are visually exaggerated; this would prevent users from making accurate distance estimations. An improved method was presented later by Glander and Döllner (2009) that allows the generalization of 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. Many studies have adopted CityGML to express geometric, topologic and semantic information of 3D city models (Reitz et al., 2009; Guercke et al., 2011). While CityGML provides great flexibility for storage and representation of 3D buildings, much of the semantic information is lost

during the conversion process (Mao et al., 2011). Fan et al. (2009) presented an approach for building generalization by CityGML that takes in consideration of semantic information associated with geometrical meshes of simplified buildings. Their approach reduces storage space, speeds up network transmission and geometric computation. Mao et al. (2011) presented a multiple representation data structure CityTree for dynamic visualization of 3D city models. Although their approach dramatically reduced the load time in visualization of 3D urban models, the visual quality will require further improvement.

A few of the approaches mentioned above for 3D urban building generalization are mainly concentrated on simplification of single buildings, but have proven difficult for rendering large urban models in real time. Others focus on simplifying city-sized collections of 3D urban buildings, but their visual quality needs to be improved. Our objective is to develop a hybrid rendering method for efficient visualization of 3D city models. Our approach will not only preserve urban legibility, but also implements dynamic visualization of 3D city models.

3. Simplification and aggregation of city building models

When we view buildings close-up, we observe that each building would carry highly detailed geometric features. We simplify the geometries of these single 3D buildings in order to keep high-quality urban models near the viewer. Meanwhile, consider a row of identical buildings separated by little space. We should combine their geometries and render far buildings as one single model. Therefore, to render large urban models quickly and preserve the visual quality of the landscape as much as possible, our method combines the single building simplification and building group aggregation to visualize urban models. Fig. 1 describes our approach for the simplification and rendering of city models.

3.1. Simplification of single 3D buildings

We simplify 3D single building models near the viewer to create different levels of detail (LODs), and then select the proper LODs to render the corresponding buildings according to their distance from the viewpoint. Our approach for simplifying these 3D single building models is divided into five phases: building footprint correction, special structure removal, roof simplification, oblique facade rectification and facade shifting.

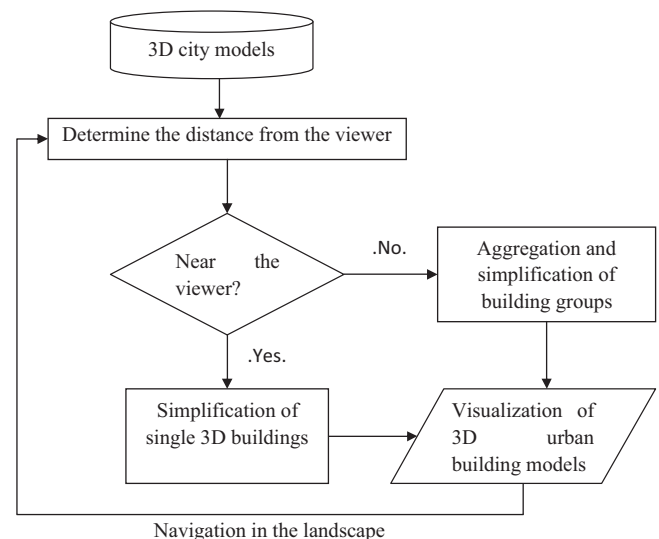


Fig. 1. Our approach uses two rendering methods, each applied in turn depending on the distance between the building and the viewer.

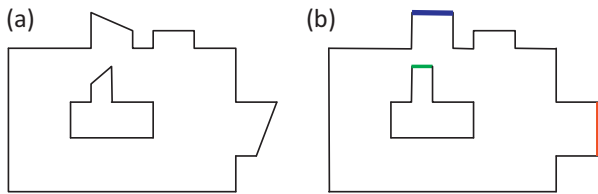


Fig. 2. Building footprint rectification. (a) Original footprint. (b) The rectified footprint. Bold and colored edges have been snapped to the same horizontal or vertical boundary.

3.1.1. Building footprint correction

Before simplifying a complex building, we need to orthogonalize the building. To achieve this, we may reorient the building walls to the principal orientations of the building footprint (see Fig. 2). The process is similar to the method of Grabler et al. (2008).

3.1.2. Special structure removal

Some building surfaces are special with tower roofs, chimneys, small pyramids or any other polyhedrons which show the building models in details. Those objects usually share a common feature: their volumes or facade areas are small. At preprocessing, we choose the volume as an indicator to pick out such structures. When the building model is taken as input, the volume of each part is calculated. If the volume is less than a given threshold, these structures are removed. By doing this, the main structures are remained (Fig. 3(b)).

3.1.3. Roof simplification

Roof structures for 3D urban building models may be quite complex. We model the roof framework straight from the building footprint's skeleton. A complex roof can be simplified by dividing a building model into several prisms and roof combinations or by assembling simpler roofs. In the case of a hip roof that has the same slope for each roof panel, the roof simplification is performed by computing the straight skeleton of the building footprint. The computation of a gable roof with the same slopes for each panel is based on the straight skeleton too. Finally, we simplify the roof with different slopes by computing the intersection of the different roof panels (Fig. 3(c)). Moreover, a roof can be cropped by computing the intersection between the roof and the horizontal plane with the defined height. The polygons resulting in this intersection are used as the footprints to reconstruct the superimposed roofs.

3.1.4. Oblique facade rectification

The phase is to rectify building facades which are not vertical extrusions of the building's rectified footprint. We therefore iterate over the set of the building facades. When the angle formed by a facade and the footprint is located at the range $[75^\circ, 90^\circ]$, we adjust the facade and make it vertical to the footprint. We match the points of the oblique facade with the points of the footprint nearest to them, after the facade is projected onto the footprint orthographically. Their x and z coordinates are set to the counterpart of the footprint points' x and z coordinates. Their y values keep unchanged. Through this process, the oblique facades are adjusted to be vertical to the footprint (Fig. 3(d)).

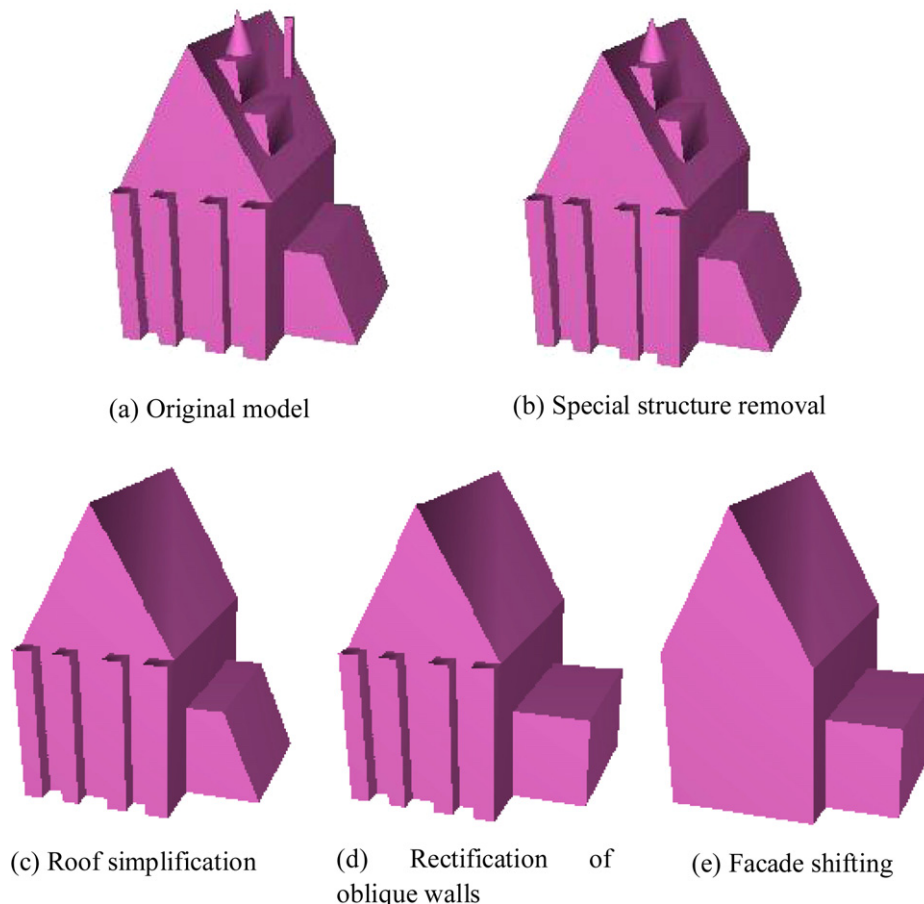


Fig. 3. The three phases of 3D building model simplification.

3.1.5. Facade displacement

Complex buildings may contain many parallel facades. Simplification can be achieved by moving parallel facades orderly. The procedure for handling this can be divided into three steps (Fig. 3). Firstly, find out the parallel facades. The angle between two facades is computed by multiplying the mesh normal vectors. Here we need to define a positive direction. The outward direction which is vertical to the facet in the counterclockwise direction is defined as the positive direction. When the dot product of the standard normal vectors of two facets is 1, they are parallel. If the dot product is -1 , we take them as non-parallel facades. An array is used to store the identifiers of the parallel facets. Some complex buildings may have several coplanar mesh groups and those groups are parallel. By dividing the array for storing the parallel façade indicators into several coplanar arrays, we can merge the coplanar façades gradually. If the distance between two façades is zero, they are coplanar. The specific process is described below: choose one point on each façade and these two points form a vector. If the dot product between this vector and normal vector of one façade is zero, they are coplanar and stored in an array. Do this step iteratively until we find out all the parallel facades from the coplanar arrays. Finally, shift two coplanar façade groups together with the minimum distance in each building. During the facet shifting process, we decide the displacement mainly based on the façade's area. Assume the distance between two groups of the coplanar facets is d , the areas of the first group and second group are $area_1$ and $area_2$ respectively. If $area_1 > area_2$, the displacement of the first group is set to be zero, and the second group is set to d , i.e. the facet group owing the larger area are not moved. The process is similar when $area_1 < area_2$. When $area_1 = area_2$, they both shift $d/2$ to merge together. We keep iterating this step until the minimum distance of two coplanar facet groups are larger than a threshold distance (Fig. 3(e)).

Table 1 shows the simplification of 3D buildings with various shapes. Table 2 shows the aggregation of 3D buildings. Our approach can handle with building models with different shapes well.

3.2. Aggregation and simplification of building groups

When the distance from the viewpoint to the urban models is larger than a distance threshold, we group these building footprints into clusters, generalize and render them in the context of Gestalt psychology and urban legibility (Yang et al., 2011). To group building footprints into clusters, we first introduce a new distance measurement method as the distance metric of the single-link clustering algorithm. Each cluster is merged based on the Delaunay triangulation and the polyline generalization algorithm. Then we construct a hierarchical tree to store these multi-resolution building models.

3.2.1. Hierarchical clusters generation

Based on Chang et al.'s (2008) single-link hierarchy method to generate building clusters, we proposed a new index, adjacent distance N , which could reflect 2D space relationship of buildings better. 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 p_1 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 p_2 , 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 Fig. 4(e)). 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 p_1 and v_j of p_2 . 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.

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, we project s_1 onto s_2 and get l_2' . Then the adjacent distance d_{near} between the polygons p_1 and p_2 can be expressed as

$$N = \frac{l_1/l_1 + l_2/l_2}{2} \quad (1)$$

$$d_{\text{near}} = d_{\min} \times (1 - N \times \lambda) \quad (2)$$

where N represents the orientation and similarity relations of the adjacent sides; λ is a weight factor, which ranges from 0 to 1. λ is used to determine the weight of N in d_{near} . The weight increases with increase in λ , which means that the relative position and directional relations should be more concerned. Empirically, $\lambda = 0.6$. 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 the adjacent sides are not parallel with each other, or have different lengths, or are staggered, d increases. Fig. 4 gives an illustration of this process.

Acquiring the adjacent sides is a key step for calculating the adjacent distance. To improve the computing efficiency, we select parts of the vertices by generating two footprints' convex hull. The convex hull can be created by the Graham scan algorithm (Graham, 1972). Through the convex hull, the number of vertices participated in calculating the adjacent sides can be reduced.

3.2.2. Generalization of building footprints

The next step is to merge two footprints into a new one, and then generalize it. In the merging procedure, our aim is to minimize the additional spaces contained by the new generated footprint. This process inevitably introduces geometric errors, because the new footprint contains previous empty spaces. The points that participated in calculating the adjacent sides are reused here to build the constrained Delaunay triangles (in Fig. 5(a), the dotted lines illustrate the generated triangles). We define a triangle pair (T_1, T_2) called a connection pair (Yang et al., 2011).

According to the definition, the triangle of a connection pair has a side which 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 Fig. 5(a), 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, this pair is called a strict connection pair. Otherwise, the pair is called as 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 is used to connect the two footprints. The new generated footprint is illustrated in Fig. 5(c). 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 Fig. 5(b) has no strict connection pairs, and the non-strict connection pair $\{v_{1-4}, v_{1-5}, v_{2-8}\}$ and $\{v_{1-5}, v_{2-6}, v_{2-8}\}$ has the minimum d_{pair} . The new generated footprint is illustrated in Fig. 5(d).

The new footprint may have too many geometric details and should be further generalized. We employ the polyline generalization algorithm proposed by Visvalingam and Whyatt

Table 1
The original building models and their corresponding simplified models.

Building type	Original building	Simplified building
L		
Regular protrusions		
Irregular protrusions		
Hollow		
Chimney on shape roof		
Chimney on symmetric roof		

(1993) to remove small concave triangles. 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 area calculation and vertex deletion are repeated until there are no concave triangles, or every concave triangle's area is larger than the threshold. Fig. 5(e)–(g) illustrate the process. The threshold is $\delta a = a_{ave}/\lambda$.

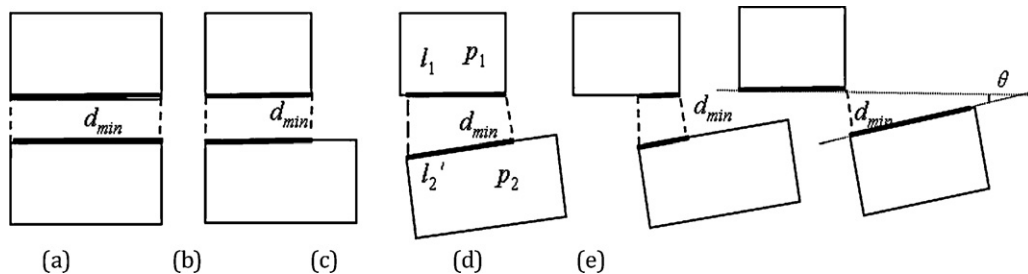
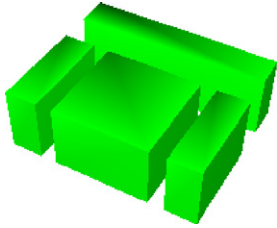
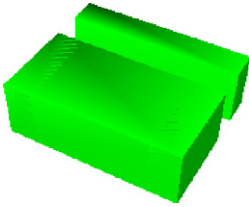
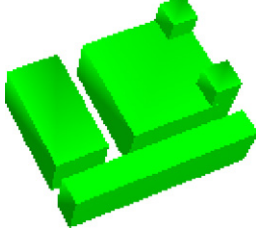
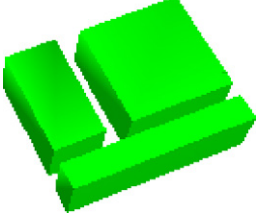
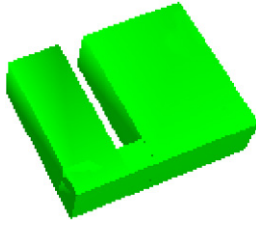
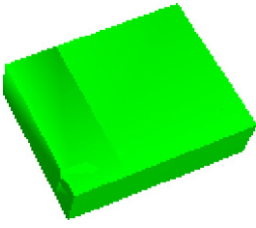


Fig. 4. The adjacent edge of the two polygons (in bold line) and minimum Euclidean distance (d_{min}). From (a) to (e), d_{min} remains the same.

Table 2
The original building models and their corresponding aggregated models.

	Original buildings	Aggregated buildings
General building models		
Gradually aggregate building models		
		

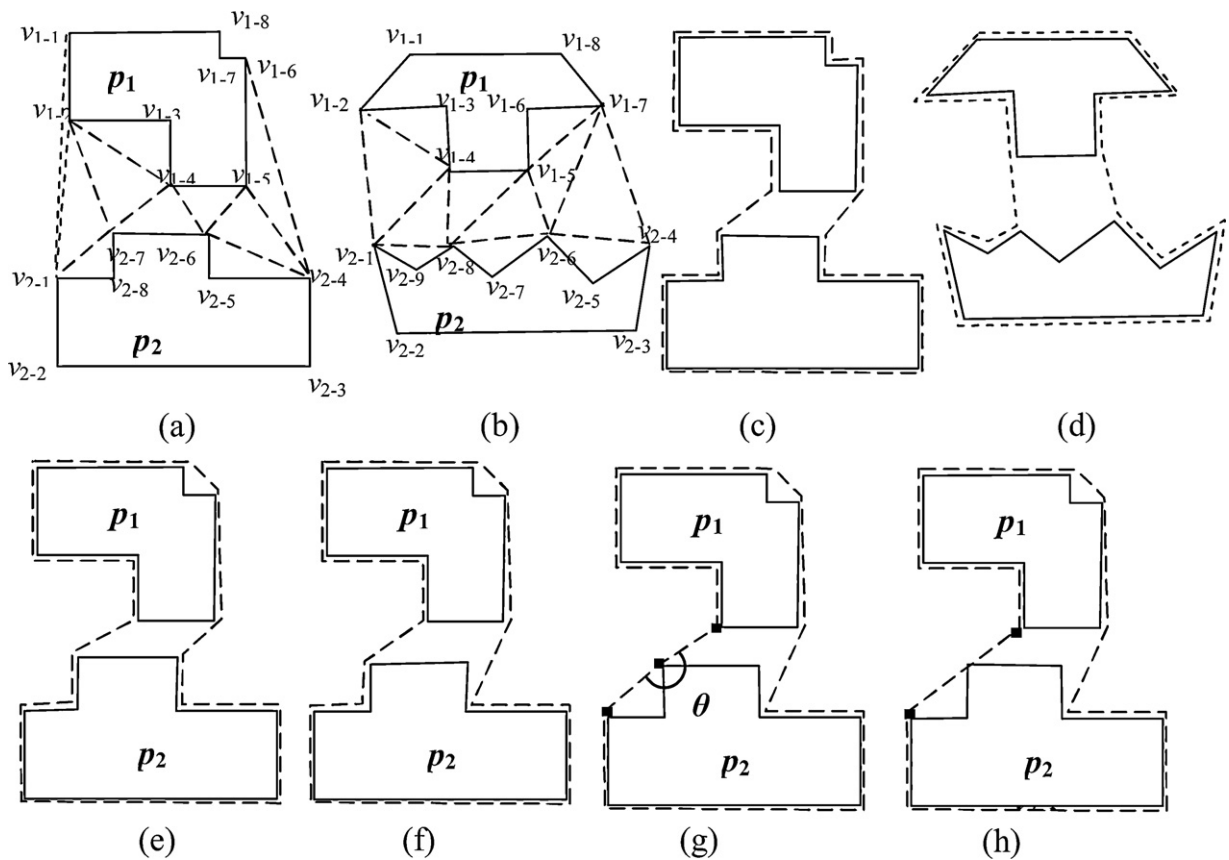


Fig. 5. Merging polygons ((a) and (b)): Original polygons and the constructed Delaunay triangulation. (c) and (d) are the results of (a) and (b), respectively.

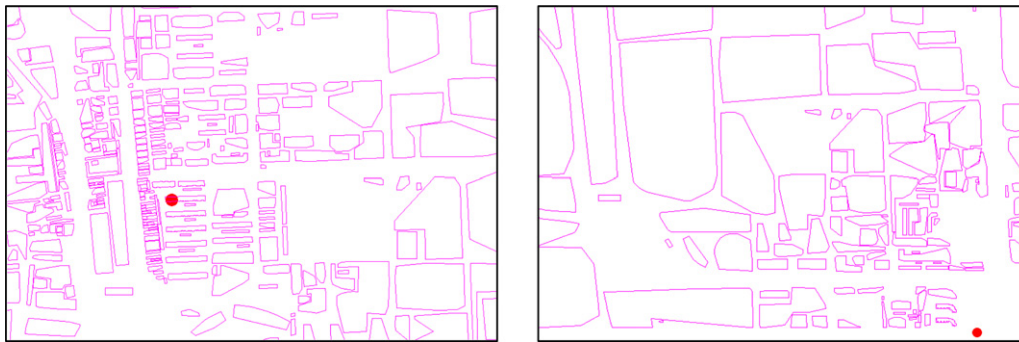


Fig. 6. Merging adjacent buildings based on the distance to the viewpoint (the red dot represents the viewpoint).

After small concave triangles are removed, the new generated footprint's vertices are traversed again to further decrease the vertices. If the angle (denoted by θ in Fig. 5g) formed by them is larger than a threshold δ (empirically, $\delta = 175^\circ$), the shared vertices by the two sides are removed. Fig. 5(g) and (h) show this process.

3.2.3. Establishment of the hierarchical tree

The hierarchical clustering process can be represented by a tree structure. Because there are no new vertices generated, the coordinates of the original vertices are stored alone. For each node in the tree, the identifiers of its corresponding vertices are stored instead of its coordinates so that we can avoid vertices' coordinates being stored redundantly and save much storage space. Attached useful information of each node is also stored, such as the footprint area and building height.

Based on the distance between the polygons defined above, the footprints with small distance can be merged step by step. As a result, adjacent buildings are merged and thus the number of buildings is declined. Fig. 6 illustrates the view-dependent simplification of the building footprints.

4. Experiments

Our approach was implemented by using Visual .Net and OpenGL applications. To validate our building rendering algorithm, experiments are performed on a Windows XP Pro machine with a 2.33 GHz, dual-core Intel Pentium III Xeon processor, 2GB RAM, and a 512 MB, Nvidia GeForce 9600GT Graphics. The building dataset of Chaoyang District, Beijing, China is used. It contains 96,821 buildings and 613,603 footprint vertices. The buildings have different shapes, such as rectangle, trapezium, L-shape, T-shape, U-shape and H-shape.

Table 3 compares the load time between the original buildings and their corresponding LOD models (the viewpoint is unchanged). Our method reduces the load time. Moreover, with

the city buildings increasing, loading the generated LOD models need much less time than loading the original buildings. The LOD creation time is not extensive compared to the total load time. During the user interaction (e.g. change of view point/zooming), the models will be reloaded to refresh the 3D scenes. It can be seen that the load time is dramatically reduced with our method.

In order to further test our method in rendering efficiency, we also record the visualization time for the original buildings and LOD models respectively. Table 4 shows the visualization time for the original models and simplified models. From Table 4, it can be seen that the visualization time is dramatically reduced with our LOD rendering method. Especially, our method has a great advantage in visualizing large city models.

Fig. 7 shows the generalized footprints by our method and Chang et al. (2008), respectively. Fig. 7(a) is a graphic illustration of the original building footprints with 85 buildings. As shown in Fig. 7, the result of (h) is similar with (e); the result of (f) can be obtained according to combination of (i) and (j) orderly. However, the wrong merging order (footprints inside red rectangle in (f) should be merged earlier than those inside green rectangle in (f)) is avoided in (f); comparing with (k), Fig. 7(g) can well extract a regular cluster border (the result inside the red rectangle in (g)), and is able to abandon complex cluster borders (the result inside the green rectangle in (g)) which greatly distorts the hull of original footprints. As a whole, both of the two methods can preserve road information efficiently. However, our generalization process can better take into account the spatial relation among adjacent buildings.

Figs. 8 and 9 show the view in the direction along the road. The red and yellow ellipses highlight the changes of the test data set. The higher height is adopted as the height of the merged buildings, as adjacent buildings have different heights. In Fig. 8, the merged buildings have facades on the same line, while in Fig. 9 the merged buildings walls are on the different lines. Fig. 8(a) and Fig. 9(a) illustrate the enveloped city models. Fig. 8(b) and (c) and Fig. 9(b) and (c) are the results of the clustering and amalgamation.

Table 3
Load time of different buildings.

Load time (s)	6325 buildings	7842 buildings	9175 buildings	10,240 buildings
Time for original buildings	31.2	38.8	42.3	49.7
Time for LOD building models	12.5	15.3	16.9	19.2
Difference value	18.7	23.5	25.4	30.5

Table 4
Visualization time of different buildings.

Visualization time (s)	6325 buildings	7842 buildings	9175 buildings	10,240 buildings
Time for original buildings	1.32	2.24	3.18	3.88
Time for LOD buildings	0.81	1.56	1.97	2.17
Difference value	0.51	0.68	1.21	1.71

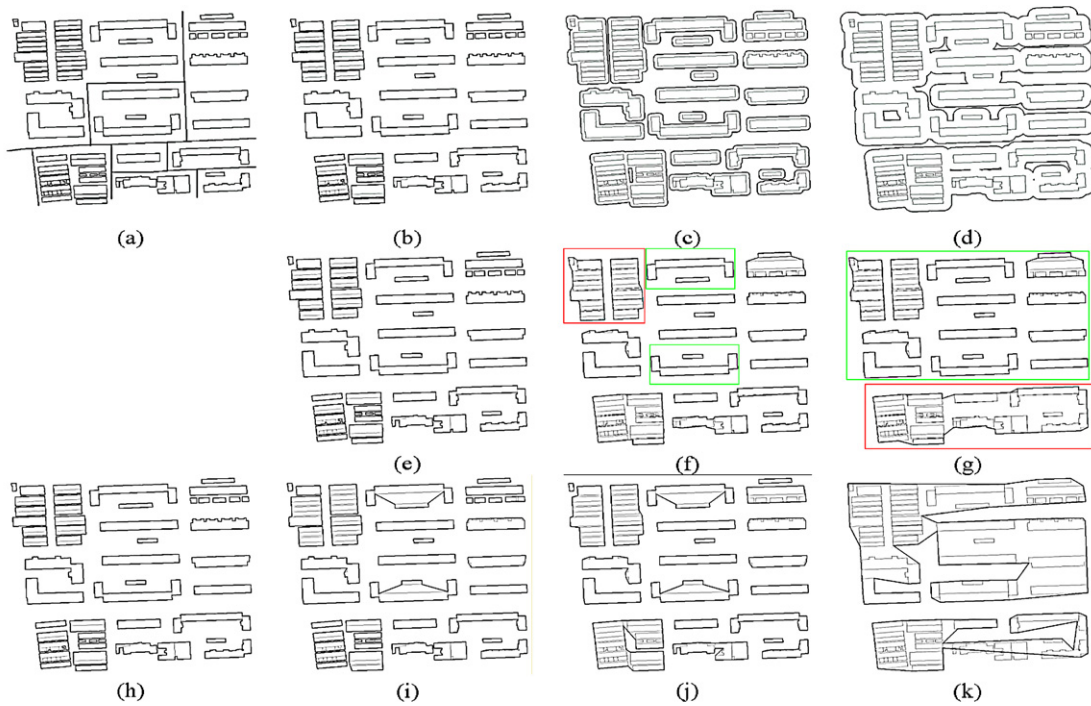


Fig. 7. Examples of the footprint generalization. (a) The original building footprints and roads, (b)–(d) show the clustering results by creating the buffers to the original footprints at increasing distances. (e), (f) and (g) are the combined results using our method, and each corresponds to clusters in (b), (c) and (d), (h)–(k) illustrate the simplified results clustered by the increasing adjacent distances improved on Euclidean distance used by Chang et al. (2008).

Due to the high density of buildings in the dataset, a good real-time visual effect can be acquired by rendering three levels of building detail. Fig. 10 illustrates a multiple

representation for dynamic visualization of 3D Beijing city models. From the figure, we can see the near building models are rendered with more details, and the details become less with the visual distance becoming far away. Building landmarks play an important role in keeping urban legibility. Buildings with distinctive visual features, like height and volume, which differ significantly from the appearance of other buildings in the local area, are likely to be remembered as landmarks (Grabler et al., 2008). From these figures, we can see that the buildings far away from the viewpoint are more generalized than the nearby ones. The buildings far away from the viewpoint are aggregated, while the near ones are not, and they have higher levels of detail.

To further validate the performance of our method, we use two approaches to render the above building dataset of Chaoyang District: rendering these models using our method and rendering the original models without simplification. Fig. 11 illustrates the performance comparisons of the two methods. Frame rates for rendering the original 3D city buildings are from 8 to 17 frames per second (fps), while those for rendering the generalized building models using our method are 45 fps on average. We see that

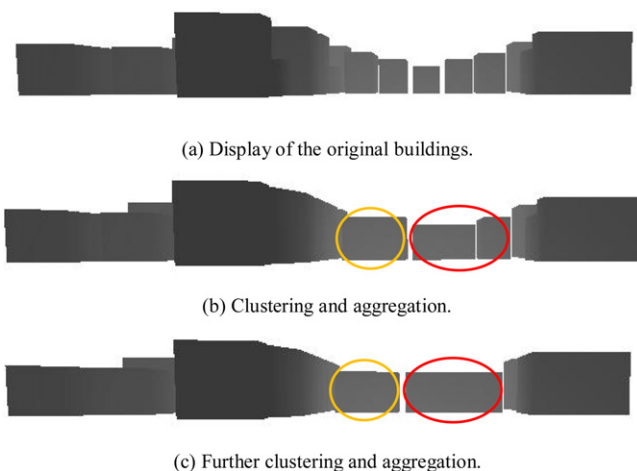


Fig. 8. Street views of 3D urban models (building facades in the same line).

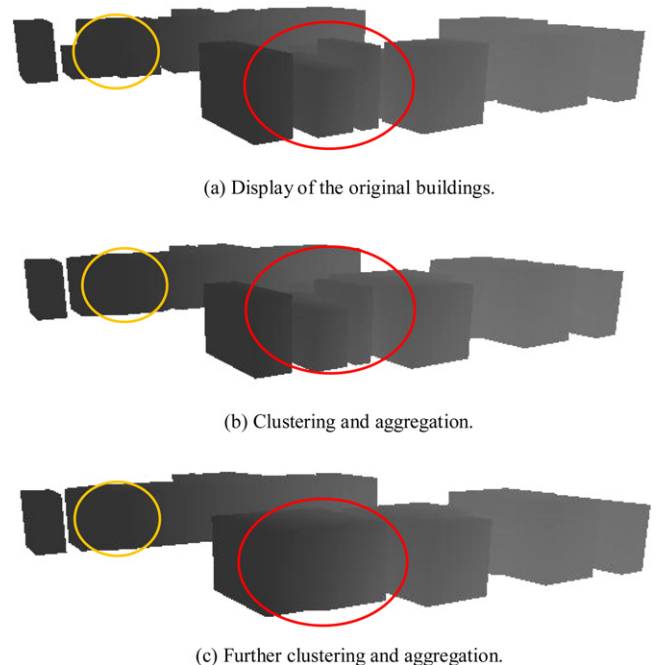


Fig. 9. Street views of 3D urban models (building facades in the different lines).

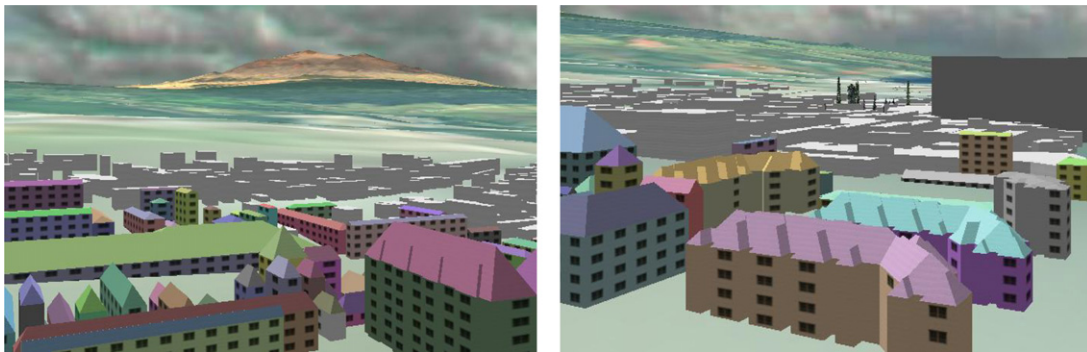


Fig. 10. Multiple representations of 3D urban models.

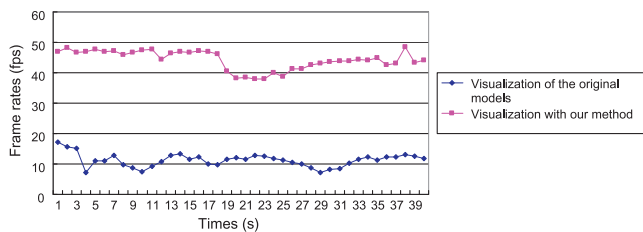


Fig. 11. Comparing the frame rates of city models visualization using our method with those of the original models visualization directly.

our proposed approach effectively accelerates the rendering process and makes a well balance between the rendering efficiency and visual quality. Meanwhile, it can maintain the districts, roads, landmarks and other elements of urban legibility, without causing visible topological errors.

5. Conclusion and outlook

This paper has addressed a new hybrid method to generalize and visualize large 3D urban models, and each applied in turn depends on the viewpoint. For the near models, 3D single buildings are simplified to create different LOD models and render them for constructing high-quality urban building landscapes. For the farther buildings, we cluster the footprints of the neighboring buildings. The clustering process is performed in the context of Gestalt psychology and urban legibility. The clustered buildings are aggregated and generalized. We then use the generalized footprints and corresponding building height to visualize far buildings. Our approach not only reduces the geometry of complex urban buildings but also successfully maintains the city's image (Lynch, 1960).

Improvement is still needed in the quality of visualization of 3D urban models. Simplifying city models coincident with human special cognition still remains an open problem. It is difficult to quantify a person's sense of spatial awareness, so more research work needs to be done for generating a humanized 3D virtual urban system. Currently, our 3D building generalization process only considers the simplification and aggregation operators. The displacement operator has to be regarded in the future as it is implied by the enlargement of important buildings.

Acknowledgements

The authors would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions. The study is supported by the National Natural Science Foundation of China

(No. 60502008) and the National 863 High-Tech Program of China (No. 2011AA120302).

References

- Aliaga, D.G., Rosen, P.A., Bekins, D.R., 2007. Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics* 13 (4), 786–797.
- Anders, K.H., 2005. Level of detail generation of 3D building groups by aggregation and typification. In: *Proceedings 22nd International Cartographic Conference*, La Coruña, Spain, pp. 9–16.
- Bai, F.W., Chen, X.Y., 2001. Generalization for 3D GIS. In: Chen, J., Chen, X.Y., Tao, C., Zhou, Q.M. (Eds.), *Proceedings of the 3rd ISPRS Workshop on Dynamic and Multi-dimensional GIS*. Bangkok, Thailand, pp. 8–11.
- Cabral, M., Lefebvre, S., Dachsbacher, C., Drettakis, G., 2008. Structure-preserving reshape for textured architectural scenes. *Computer Graphics Forum* 28, 469–480.
- Chang, R., Butkiewicz, T., Ziemkiewicz, C., Wartell, Z., Ribarsky, W., Pollard, N., 2008. Legible simplification of textured urban models. *IEEE Computer Graphics and Applications* 28 (3), 27–36.
- Fan, H.C., Meng, L.Q., Jahnke, M., 2009. Generalization of 3D buildings modelled by CityGML. *Advances in GIScience*. In: *Proceedings of the 12th AGILE Conference*, Hannover, Germany, pp. 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., 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., 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*, Montpellier, France.
- Glander, T., 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*.
- Grabler, F., Agrawala, M., Sumner, R.W., Pauly, M., 2008. Automatic generation of tourist maps. *ACM Transactions on Graphics* 27 (3), 11.
- Guercke, R., Götzelmann, T., Brenner, C., Sester, M., 2011. Aggregation of LoD 1 building models as an optimization problem. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (2), 209–222.
- Kada, M., 2005. 3D building generalisation. In: *Proceedings of the 22th International Cartographic Conference*, La Coruna, Spain.
- Kada, M., 2007. 3D building generalisation by roof simplification and typification. In: *Proceedings of the 23th International Cartographic Conference*, Moscow, Russian Federation.
- Kada, M., 2011. Aggregation of 3D Buildings using a hybrid data approach. *Cartography and Geographic Information Science* 38 (2), 154–161.
- Lynch, K., 1960. *The Image of the City*. Cambridge/MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, and London, England, pp. 5–8.
- Mao, B., Ban, Y.F., Harrie, L., 2011. A multiple representation data structure for dynamic visualisation of generalised 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (2), 198–208.
- Meng, L.Q., Forberg, A., 2007. 3D building generalisation. In: Mackaness, W., Ruas, A., Sarjakoski, T. (Eds.), *Challenges in the Portrayal of Geographic Information*. Elsevier Science, Amsterdam.
- Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64 (6), 575–584.
- Reitz, T., Kramer, M., Thum, S., 2009. A processing pipeline for X3D earth-based spatial data view services. In: *Proc. 14th International Conference on 3D Web Technology*, Darmstadt, Germany, 16–17 June, pp. 137–145.

- Royan, J., Balter, R., 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, pp. 432–439.
- 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).
- Thiemann, F., Sester, M., 2006. 3D-Symbolization using adaptive templates. ISPRS Technical Commission II Symposium, Vienna.
- Visvalingam, M., Whyatt, J.D., 1993. Line generalisation by repeated elimination of points. *The Cartographic Journal* 30 (1), 46–51.
- Yang, L., Zhang, L., Ma, J., Xie, J., Liu, L., 2011. Interactive visualization of multi-resolution urban building models considering spatial cognition. *International Journal of Geographical Information Science* 25 (1), 5–24.