

Article

# Texture-Cognition-Based 3D Building Model Generalization

Po Liu <sup>1</sup>, Chengming Li <sup>1,\*</sup> and Fei Li <sup>2</sup>

<sup>1</sup> Chinese Academy of Surveying & Mapping, Beijing 100830, China; liuposwust@163.com

<sup>2</sup> Key Laboratory of Watershed Geographic Sciences, Nanjing Institute of Geography and Limnology, Chinese Academy of Sciences, Nanjing 210008, China; lfgis@163.com

\* Correspondence: cml@casm.ac.cn; Tel./Fax: +86-010-6388-0555

Academic Editors: Jianming Liang, Jianhua Gong, Yu Liu and Wolfgang Kainz

Received: 28 May 2017; Accepted: 21 August 2017; Published: 23 August 2017

**Abstract:** Three-dimensional (3D) building models have been widely used in the fields of urban planning, navigation and virtual geographic environments. These models incorporate many details to address the complexities of urban environments. Level-of-detail (LOD) technology is commonly used to model progressive transmission and visualization. These detailed groups of models can be replaced by a single model using generalization. In this paper, the texture features are first introduced into the generalization process, and a self-organizing mapping (SOM)-based algorithm is used for texture classification. In addition, a new cognition-based hierarchical algorithm is proposed for model-group clustering. First, a constrained Delaunay triangulation (CDT) is constructed using the footprints of building models that are segmented by a road network, and a preliminary proximity graph is extracted from the CDT by visibility analysis. Second, the graph is further segmented by the texture-feature and landmark models. Third, a minimum support tree (MST) is created from the segmented graph, and the final groups are obtained by linear detection and discrete-model conflation. Finally, these groups are conflated using small-triangle removal while preserving the original textures. The experimental results demonstrate the effectiveness of this algorithm.

**Keywords:** 3D building generalization; texture; SOM; cognition; constrained Delaunay triangulation

## 1. Introduction

Virtual Geographic Environments (VGEs) have been proposed as a new generation of geographic analysis tools to improve our understanding of the geographic world and assist in solving geographic problems at a deeper level [1,2]. Cities are the centers of living; people mainly work and live in cities; three-dimensional urban-building models are key elements of cities that have been widely applied in urban planning, navigation, virtual geographic environments and other fields [3–8]. Building visualization is the basis for collaborative work, planning and decision making in city VGEs. These models include many details to address the complexities of urban environments; thus, level-of-detail (LOD) technology is often used to reduce the amount of data and increase the rendering speed [9]. These detailed models can be replaced by less-detailed models from a more distant view using generalization [10]. Two different tasks exist for building generalization: simplification and generalization [11]. Single-building simplification has been widely studied [10,12,13], but few works have considered model-group generalization. This paper focuses on model-group generalization.

For large-scale city models, single-building simplification alone does not suffice for visualization. The huge number of single objects contributes most to both the computational complexity and cognitive load. Building LOD models can significantly assist the interactive rendering of massive urban-building models. Therefore, we must apply the principles of generalization to their visualization to efficiently communicate spatial information [14]. A building-generalization technique for linear building groups

was created by [15]. Chang et al. [16] proposed a single link-based clustering method to group city-sized collections of 2.5D buildings while maintaining the legibility of the city [17]. Glander and Döllner [18] used an infrastructure network to group building models and replace them with cell blocks while preserving local landmarks. Kada [19] extended cell decomposition to use morphological operations on a raster representation of the initial vector data to generalize building groups. Yang, Zhang, Ma, Xie and Liu [14] proposed a new distance measurement to improve the single-link clustering algorithm that was proposed by Chang, Butkiewicz, Ziemkiewicz, Wartell, Ribarsky and Pollard [16]. Guercke et al. [20] expressed different aspects of the aggregation of building models in the form of mixed-integer-programming problems. He et al. [21] also proposed a footprint-based generalization approach for 3D building groups in the context of city visualization by converting 3D generalization tasks into 2D issues via buildings' footprints while reducing both the geometric complexity and information density. Zhang et al. [22] also proposed a spatial cognition analysis technique for building displacement and deformation in city maps based on Gestalt and urban-morphology principles.

These approaches are mainly based on geometric characteristics. However, several problems remain unresolved. (1) Texture features have not yet been fully considered for generalization. Compared to geometric features, texture features more closely reflect people's visual perception. (2) Landmarks play an important role in cities and thus should be preserved during generalization. (3) Gestalt clustering in local regions is still a difficult problem. In addition, the discrete models after segmentation must be further combined, and detailed textures should be preserved in different LOD models.

This paper is the first to introduce texture features into 3D-model-group clustering and generalization based on Gestalt and urban-morphology principles. A self-organizing mapping (SOM)-based classification approach is proposed for texture-feature analysis. In addition, a new hierarchical-clustering method is proposed. First, a constrained Delaunay triangulation (CDT) is constructed from the footprints of models and other urban morphologies that are segmented by the road network. Then, the preliminary graph is extracted from the CDT by visibility analysis, the proximity graph is further segmented by texture analysis and landmark detection, and the final group is obtained by Gestalt-based local-structure analysis. Finally, these groups of models are conflated using a small-triangle-removal strategy.

The remainder of this paper is organized as follows. Section 2 describes the related work, namely, cognition-based 3D-model generalization and texture-feature analysis. Section 3 introduces the framework of the algorithm. A cognition-based hierarchical generalization and an SOM-based texture-classification approach are described in detail. Section 4 shows the experiment. Finally, the study's results are summarized and discussed in Section 5.

## 2. Related Work

This section discusses building-model generalization and texture feature.

### 2.1. Cognition-Based 3D-Building Generalization

A large number of simplification algorithms that are based on different perspectives in computer graphics have been proposed. Garland and Heckbert [23] developed the first simplification method based on quadric matrices to produce high-quality approximations of polygonal models. An improved quadric error metric for simplifying meshes was further proposed based on geometric correspondence in 3D to preserve the color, texture and normal attributes [24,25]. Lindstrom and Turk [26] introduced a framework that uses images to determine the portions of a model that must be simplified. High-quality textured surfaces are produced with the edge-collapse algorithm. These works are mainly based on single buildings. As an exception, Chang, Butkiewicz, Ziemkiewicz, Wartell, Ribarsky and Pollard [16] proposed a single link-based clustering method to group city-sized collections of 2.5D buildings while maintaining the legibility of the city, and the texture for each face was generated by placing an orthographic camera so that its near clipping plane lay on the face.

Buildings are the main content of map generalization, and structural characteristics are the main purpose of generalization. The urban-morphology theory and Gestalt principles are used to form global and local constraints during building-model generalization, respectively [27]. Regnaud [28] detected and organized building relationships for 2D map generalization using a minimum support tree (MST) and a proximity graph of the building set according to various criteria from Gestalt theory. Li et al. [29] presented a cognition method that combines an urban morphology and Gestalt theory for 2D-building groupings and generalizations, but this method is difficult to apply to complex building shapes. Yan et al. (2008) improved this method for buildings, but the map generalization efficiency is still rather low.

The existing map-generalization method can be extended to 3D-model generalization. Global groups are suitable for urban-morphology theory. Mao et al. [30] used an urban-road network to segment models while maintaining the significant city layout. Yang, Zhang, Ma, Xie and Liu [14] proposed an approach that is based on the single-linking clustering method to generalize and render urban-building models while maintaining their main characteristics, such as neighborhoods, paths, and intersections [17]. Gestalt theory can also be used to cluster buildings. Mao et al. [31] proposed a linear group-detection method for building classification based on MST, but this method is not suitable for complex building models in urban environments. Zhang, Hao, Dong and Zhen [22] proposed a support vector machine (SVM)-based method to group models according to their proximity, similarity, and direction. Local constraints from Gestalt theory and global constraints from urban morphology have also been used to update and visualize groups of 3D models [32,33]. These generalization methods are based on geometric characteristics and rarely consider the texture. Texture characteristics are used to assist generalization in this paper based on this previous cognition-based 3D-building-generalization approach.

## 2.2. Texture Feature

Texture is typically defined as an image of a local feature or an indicator of pixels in the local region. Texture is a value for the characteristics of internal variance and is associated with the location of the texture, size, and shape but is not related to the brightness. Structural texture-space differences can be converted into different gray values and expressed by a mathematical model. The texture itself contains the direction, position and shape of these characteristics, which can be described by the geometrical characteristics of the model. Texture-analysis methods include both statistical and structural analysis. Statistical characteristics include the texture area of the gray histogram, the gray mean and variance, and the minimum and maximum pixel values. Structural analysis is similar to the geometric analysis of building footprints and is highly suitable for artificial textures, such as buildings. Structural analysis is treated as a geometric clustering problem in this paper.

Current 3D-model generalization is mainly based on geometric features because texture information has still not received sufficient attention. Unlike other methods, Noskov and Doytsher [11] proposed a generalization method that is based on the rasterization of the 2D footprints of 3D buildings, but the pixel characteristics still do not include the texture. The texture can be used as an important classification indicator during generalization, which is more in line with visual-cognitive habits. Cabral et al. [34] proposed a geometry- and texture-matching method to reshape texture models that can be applied to an existing model of a small grid but is not suitable for a large-scale 3D-city model. Zhang et al. [35] realized the unified management of the texture and geometric structure but did not make full use of texture information. In this paper, the texture features of a 3D model are combined with geometric information for model generalization. The texture must occasionally be reshaped after changing the geometry of the model groups [16,18,36] because the original textures are often eliminated during model conflation; this task is also addressed in this paper through the triangle-removal strategy.

### 3. Cognition-Based Hierarchical Clustering and Generalization

First, the texture is introduced based on urban morphology and Gestalt theory to assist the group clustering and generalization of 3D-building models, and the detailed textures are preserved after conflation. Figure 1 shows the main framework of the algorithm, which includes six steps: (1) data preprocessing: extract the footprint and texture of the model; (2) urban morphology-based global clustering: CDT construction from building footprints, the segmentation of roads according to urban-morphology theory, and the extraction of the initial proximity graph by visibility analysis; (3) the further segmentation of the proximity graph by texture analysis and landmark detection, which is obviously different from the 2D generalization method; (4) Gestalt theory-based local clustering: the generation of an MST from the segmented graph based on the nearest distance, followed by linear detection and discrete-polygon conflation, which originate from Gestalt theory; (5) the conflation of the grouping models and regeneration of the texture; and (6) the visualization of these models.

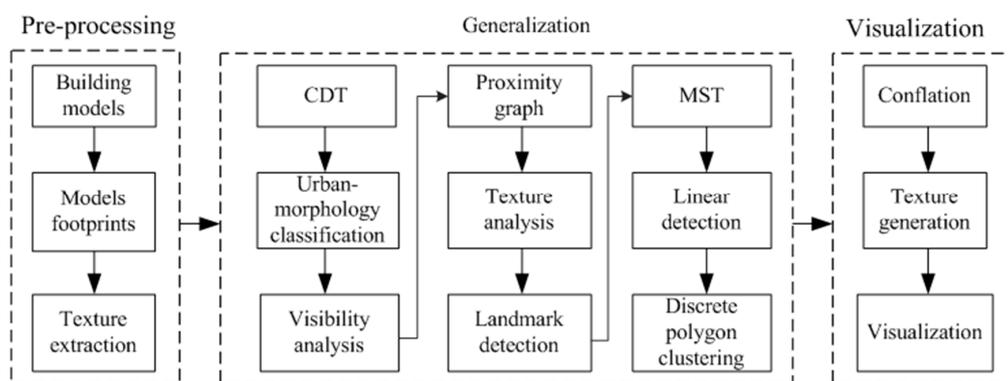


Figure 1. Algorithm framework.

#### 3.1. Data Processing

First, a 3D model must be preprocessed using the following steps: (1) extract the footprint: the model should be projected in a horizontal direction and then merged into a polygon [37]; the GEOS DLLs can be used for polygon merging; (2) calculate the centric coordinates of the footprints, and obtain the building height; the maximum Z of the vertex is taken as the building height; and (3) extract the roof and wall of the model from the normal of the triangles; the roof and facade textures can also be extracted from the original models. Figure 2a shows a visualization effect of the original model with detailed textures; the green line below the model marks the footprints that were extracted from the original model. Figure 2b presents its texture image.

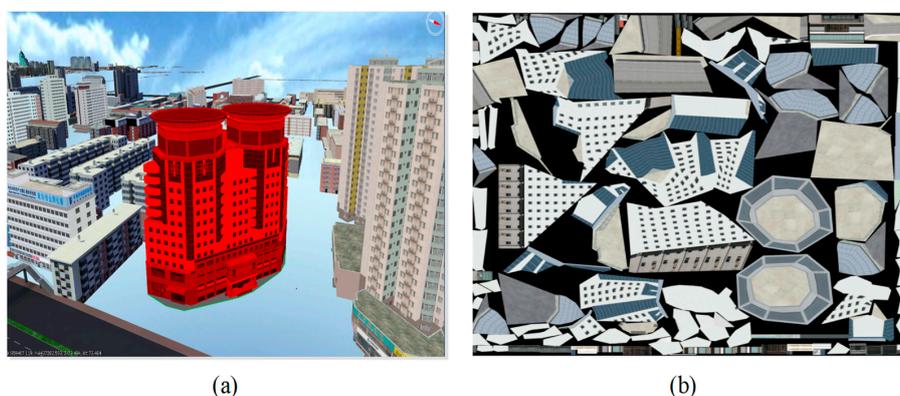


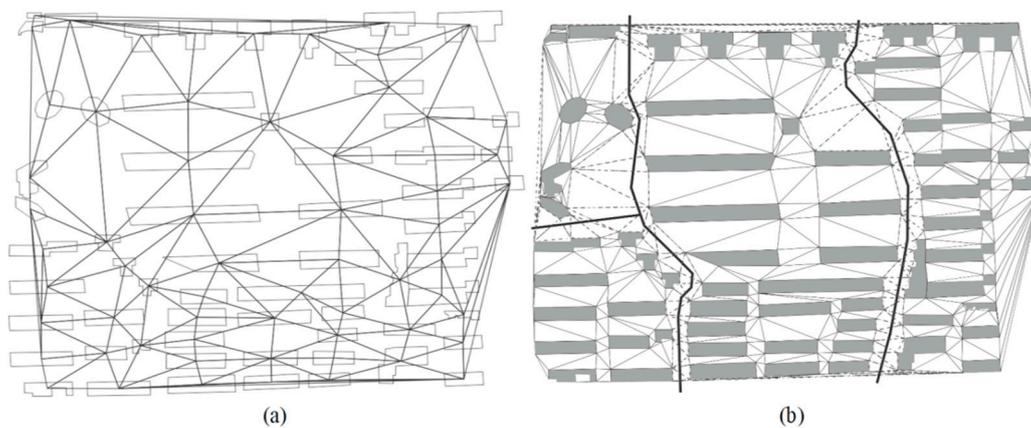
Figure 2. Experimental data: (a) original model-visualization effect with footprints; (b) texture image of the selected model.

### 3.2. Urban Morphological-Based Global Clustering

We select a typical region from the research data, which has 63 buildings, to better explain the process and effect of the algorithm; the algorithm is described with these polygons in this section. In this paper, the building models are preliminarily clustered by the urban-morphological theory, and proximity graphs are used, which include the following two steps: (1) a CDT is generated by the building footprints and further segmented by the road network; and (2) a proximity graph is extracted from the CDT via visual analysis.

#### 3.2.1. CDT

Proximity graphs are widely used in map or model generalization [22,28,29]. Zhang, Hao, Dong and Zhen [22] constructed an MST from an improved Delaunay triangulation, in which the vertices of the triangles are often located at the centers of the buildings (Figure 3a). However, the adjacent polygons within the visual scope are only directly consolidated during model merging. In this paper, we construct a constraint Delaunay triangulation based on the footprints of the 2.5D models (Figure 3b), and the triangles in the polygons are removed.



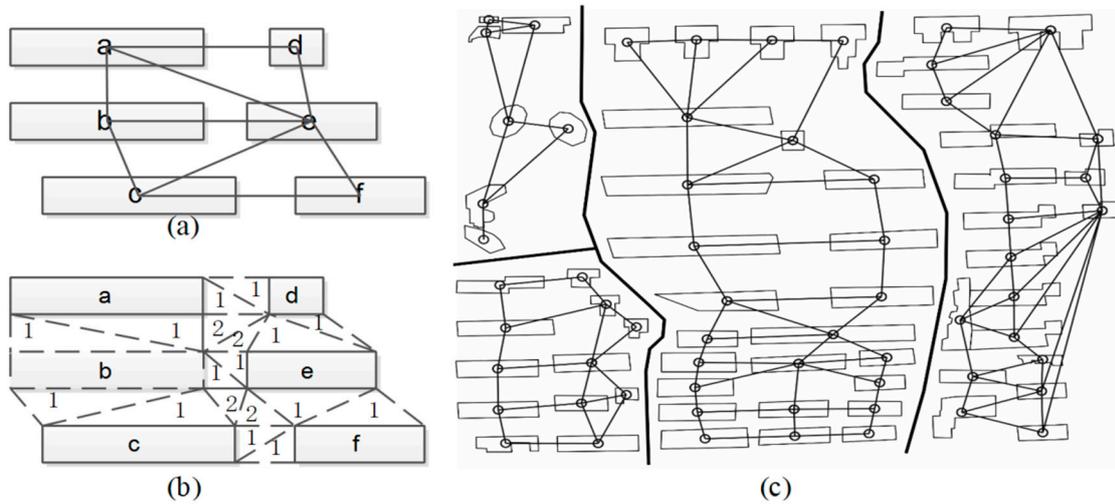
**Figure 3.** Delaunay triangulation: (a) non-constraint; (b) boundary constraint.

This Delaunay triangulation must be further segmented based on urban-morphology theory. In this paper, roads are used to divide these triangles, and the triangles that intersect with roads are removed. In Figure 3b, the triangles that intersect with roads are represented by dotted lines, while the remaining triangles conform to the urban-morphology distribution. When no road data were available, the models could be preliminarily segmented by the buffer-like image-based method from Zhang, Zhang, Takis Mathiopoulos, Xie, Ding and Wang [35].

#### 3.2.2. Visibility Analysis

We used CDT to detect the topological adjacency relationships of buildings and generated two-building groups. When the triangles were segmented, the proximity graph was further extracted by visibility analysis. Figure 4a shows a portion of the Delaunay triangulations in Figure 3a. Polygons *a* and *e* and polygons *e* and *c* could not be directly combined in the generalization, so the edges in Figure 4a do not represent the likely conflation relationship. The corresponding CDT is represented by dotted lines in Figure 4b. These triangles can be divided into two catalogues. The first category has one/two vertices on one polygon and the other two/one vertices on another polygon. The second category, which connects three polygons, was removed. Finally, if two buildings still had linking triangles, then they were connected by a line, and the vertex of the line was on the gravity of the building. Figure 4c presents the initial proximity graph from visibility analysis. Each edge shows the

directly merging relationship of two adjacent buildings. Subsequent classifications were all based on this proximity graph.



**Figure 4.** Visibility analysis: (a) Delaunay triangulation (DT); (b) constrained Delaunay triangulation (CDT); (c) preliminary proximity graph.

### 3.3. SOM-Based Texture Classification and Analysis

Texture is an unquantifiable feature; the demarcation between texture features is often unclear, although it is intuitive in human vision. Thus, an intelligent algorithm is a realistic choice. To increase the degree of automation of the algorithm, we used a SOM-based method to automatically classify texture in this paper, with the appropriate classification indicators and strategies carefully selected.

#### 3.3.1. Texture-Feature Analysis

Texture has color and image characteristics. These features reflect both the structure of gray information and the spatial information between the texture image’s gray-level distributions. 3D-building models mainly include facade and roof textures; only roof textures were analyzed in this paper. Building models in the same region often have similar characteristics because buildings that are constructed at the same time may have the same texture.

We assumed that the set of planes for a building mesh is defined as  $f(v) = \{(a, b, c, d) | ax + by + cz + d = 0\}$ . The roof and facade planes were initially clustered based on the normal of the plane. If we assume one plane  $F$ , the zenith angle  $\theta$  of the normal is defined as

$$\theta = \arccos \left\{ \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right\}. \tag{1}$$

If the angle is less than 10 degrees, then this plane is catalogued as a roof. The characteristics of the texture pixels can be evaluated by texture mapping. As shown in Figure 5, triangle  $a(V_4V_5V_6)$  has three vertices in the model mesh, and triangle  $b(t_4t_5t_6)$  is its corresponding triangles in the texture image. If we assume that the width and height of the texture image is  $w$  and  $h$ , the texture coordinates of  $V_4$  are  $(x, y)$ , where  $x$  and  $y$  are between 0 and 1; then, the coordinate of  $t_4 (t_x, t_y)$  in Figure 5c is

$$\begin{matrix} t_x = w * x \\ t_y = h * y \end{matrix} \tag{2}$$

The pixels in the triangle can be computed by an intersection operation, and the mean and variance value of triangle a are defined as

$$\bar{p} = \frac{\sum_{i=1}^n p_i}{n} \tag{3}$$

$$\delta = \frac{\sqrt{\sum_{i=1}^n (p_i - \bar{p})^2}}{n} \tag{4}$$

where  $n$  is the number of pixels inside triangle b, and  $p_i$  is the pixel value.

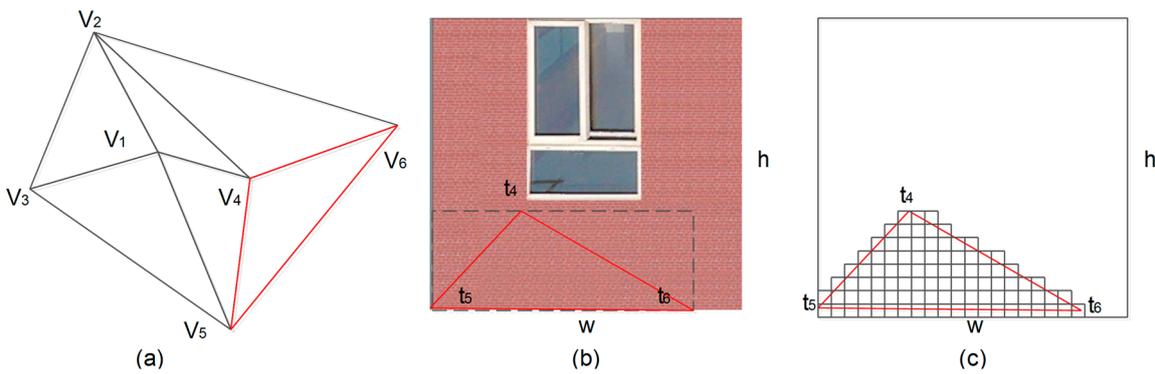


Figure 5. Texture mapping: (a) original triangle; (b) texture image; (c) pixel in the triangles.

Figure 6 shows the buildings with detailed textures in cities. Distinguishing buildings 5 and 7 from surrounding buildings (such as buildings 1, 2, 3, 4, 5 and 6) is difficult when using only the geometric features, such as the area, direction and height index, but this process is easier when using the texture.

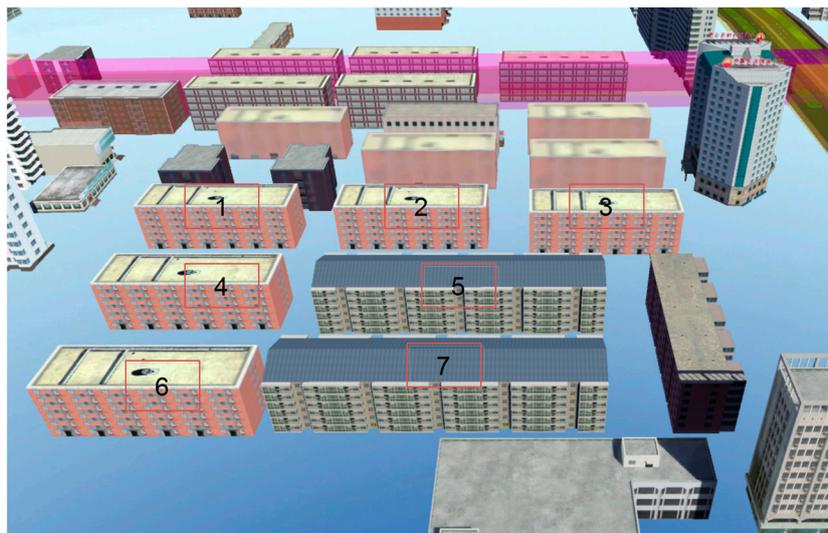


Figure 6. Texture features.

### 3.3.2. Texture Classification

A statistical texture-analysis method was used in this paper, and the mean value and variance of the texture pixels were used as two classification indices. The pixels of textures are represented by a

color. The RGB color values must be converted to HSV values before classification, and the tone is an input value of the training. Figure 7 shows the clustering results from the mean value of the pixels; the numbers in the polygon represent the identification of the buildings, and the blue lines represent roads. Distinguishing buildings 51-2 and 50-4 is difficult, but they can be easily distinguished if the variance index is added as an indicator. The numbers 51 and 50 are the FIDs of the buildings, whereas 2 and 4 are the corresponding texture-classification categories.



Figure 7. Texture classification by statistical analysis.

Classification in local regions can also significantly improve the accuracy of the SOM-clustering method. Figure 8a,b present the texture-clustering results in the global and local regions. Buildings 46 and 47 in Figure 8a are divided into different categories but belong to the same category in Figure 8b. We often demarcate the texture with surrounding buildings; therefore, texture classification in the global region is not a good choice. The local regions can be obtained from urban-morphology clustering.

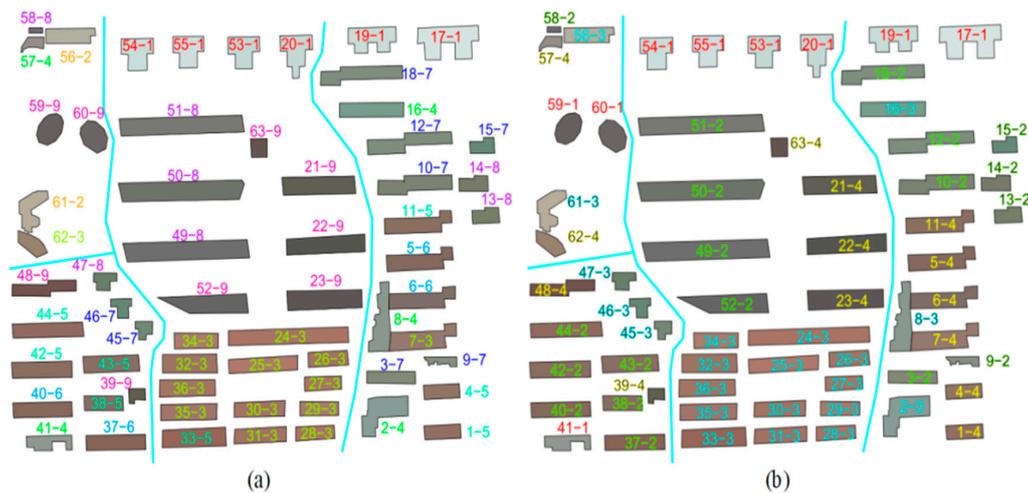


Figure 8. Texture classification in the (a) global region and (b) local region.

Building-texture features imply the structural characteristics of the 3D model, and buildings that are clustered by texture more closely match human-vision characteristics. Figure 9 shows the final proximity graph, which is segmented by the types of textures.



**Figure 9.** Proximity graph that is segmented by the types of textures.

### 3.4. Landmark Detection

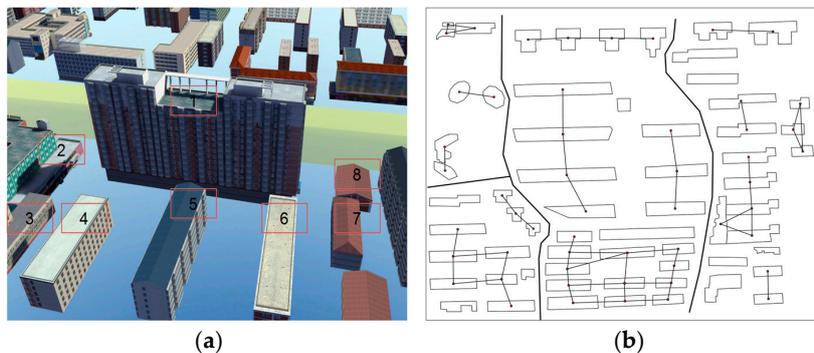
Landmarks are important components of a city because they distinguish the city from others considerably during generalization [16], and the sum of mesh planes is often complex [17]. Landmarks are widely used in navigation and must be preserved during model generation [14]. Landmark models can be detected from surrounding models by calculating differences in the height, area and volume [38]. Of these geometric features, the height of a building is extremely important and must be carefully considered during generalization [16].

Here, we demonstrate height detection. The height of building  $B_1$  in Figure 10 is  $h$ , and the surrounding buildings  $B_2, B_3, B_4, B_5, B_6, B_7$  and  $B_8$  in the visual region have corresponding heights of  $h_i$ . The average height of the surrounding buildings is

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

$$\delta = |h - \bar{h}|$$

The  $\delta$  of each model can be calculated. If the height is greater than a certain threshold, then the building can be considered a landmark. The threshold can be obtained with data training; the default value is 10 m. The heights of buildings 63 and 24 (in Figure 7) are significantly different from those of the surrounding buildings and thus can be easily distinguished through height detection, although other geometric features (such as the area) can also be used. Figure 10b shows the proximity graph, which is segmented by landmarks.



**Figure 10.** Landmark detection: (a) proximity relationship; (b) graph from landmark detection.

### 3.5. Gestalt Theory-Based Local Clustering

If the nearest building is not suitable for merging during generalization, other characteristics, such as the direction or spacing, should be considered. Three principles of Gestalt theories, namely, proximity, similarity, and common direction, were considered to extract potential Gestalt-building clusters in local regions. Texture features also include geometric-structure characteristics, and the proximity graph requires further structure analysis. In this paper, the structure analysis included three steps: MST construction, linear detection and discrete-polygon conflation.

#### 3.5.1. MST

MSTs are a powerful support tool for spatial clustering. MSTs are a polygon group of adjacent graph connections, including all interlinking points, non-closed rings, connection-edge distances and minimum characteristics of the tree. In this paper, MSTs were extracted from the proximity graph, which was segmented through texture and landmark detection, based on the nearest distance. Figure 11 shows the MST of the proximity graph in Figure 10b: its vertices are the building's center of gravity, and the edge is the distance between the adjacent buildings.



**Figure 11.** Minimum support tree (MST).

#### 3.5.2. Linear Detection

In visual perception, we tend to merge buildings in the collinear direction [39], and MSTs require further linear detection. Figure 11a shows the preliminary proximity graph. Figure 12b shows the corresponding MST, and the alphabet in the polygon shows the gravity of the building. Linear detection involved three steps. (1) From the starting point  $a$  of the MST (Figure 12b),  $ab$  is the starting edge, and  $bc$  is the next edge. If the direction difference of  $ab$  and  $bc$  is below a certain threshold, remove  $bc$ . (2) Look up the initial proximity graph (in Figure 8b) and find the edge  $bc$  that links point  $b$ . If the direction difference between  $ba$  and  $ab$  is below a certain threshold, then link  $bc$ . (3) After processing polygon  $b$ , repeat the first and second steps for polygon  $c$  until the entire tree is traversed. Figure 12c is the final proximity graph from linear detection.



Figure 12. Linear detection: (a) preliminary proximity graph; (b) MST; (c) final graph.

### 3.5.3. Discrete-Polygon Clustering

After linear detection, any discrete polygons must be further processed [29]. Compared to the preliminary graph, discrete polygons can be combined with the existing clustering groups according to the orientation, height and area features. Figure 13a shows the discrete polygons *g* and *h*; *g* may link to polygons *e* and *f* in the preliminary graph. If the direction difference between *he* and *eb* is below a certain threshold, then polygons *h* and *e* can be linked, and polygons *g* and *e* are also linked by linear detection. Figure 13b is the final proximity graph from discrete-polygon clustering.

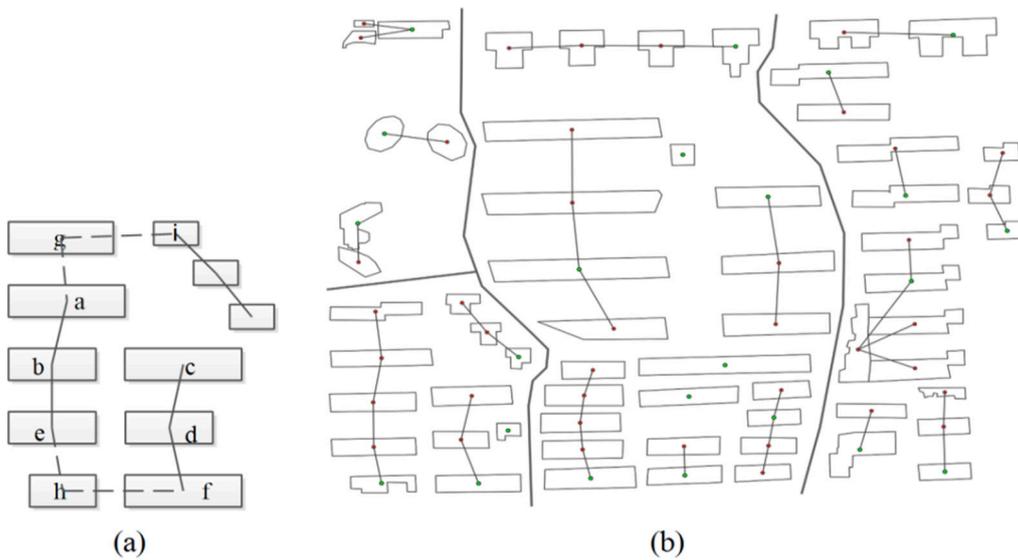


Figure 13. Discrete-model clustering: (a) discrete-polygon linking; (b) the final proximity graph.

### 3.6. Conflation

Building models must be further conflated after clustering. Yang et al. (2011) used Delaunay triangulation for 3D-model simplification and conflation; however, this approach is not practical for a detailed texture model. When the vertices are changed, we must merge the original textures. A simple model-conflation algorithm is proposed in this paper to merge the original textures. When the models

belong to the same group, they are conflated and simplified by the area threshold. The main process is as follows:

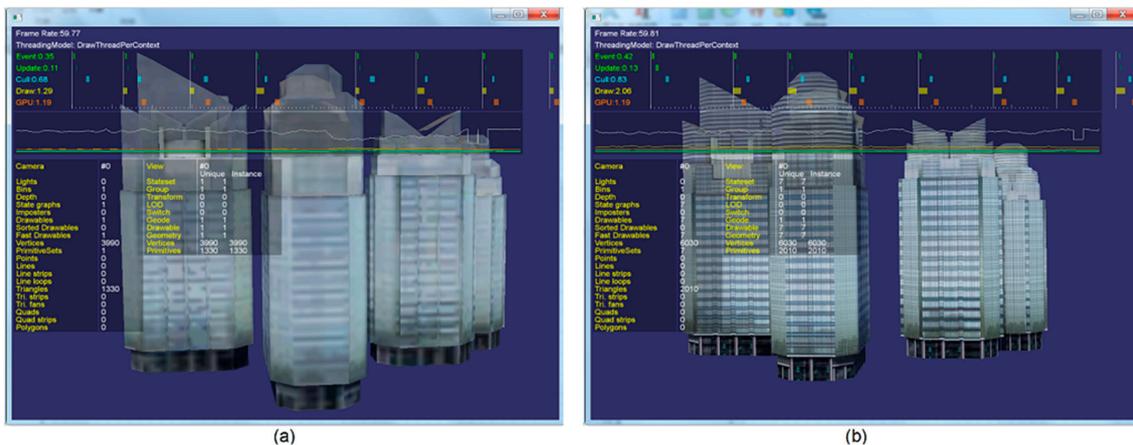
- (1) Compute the area of every triangle and sort the triangles by area.
- (2) Compute the average area of the triangles in the group and calculate the area threshold as follows:

$$\Delta s = \delta * \frac{\sum_{i=1}^n S_i}{n} \quad (6)$$

where  $S_i$  is the area of the 3D model, and  $\delta$  is the multiplication coefficient.

- (3) Traverse all the triangles, and remove any triangles whose area is below the threshold.
- (4) Conflate the remaining triangles in the group and compress their textures.

Figure 14a shows the conflated and simplified building models, while the original group models are shown in Figure 14b. As is shown in the Figure 14, the original models contained 2010 triangles with 6030 vertices; after conflation (with  $\delta$  set to 0.2), the models contained 1330 triangles with 3990 vertices. The texture and vertex coordinates of the conflation models were the same as in the original model because of the triangle-removal process. The threshold of  $\delta$  can be achieved by experience, or tested with existing models, the main feature must be preserved after triangles are removed.



**Figure 14.** Visualization of model conflation: (a) conflation model; (b) original model.

### 3.7. Model Visualization

We used a tree structure to store the simplified models; these data were discussed in detail by [14]. We only generated a new mesh, and the original textures were all preserved. As shown in Figure 15a, the original model footprints a, b, c, d, e, and f were sequentially clustered in the order that is shown in Figure 15b. The nodes that were generated by the previous steps could be merged into one node by model conflation. In the real world, the distances between adjacent buildings in the same district are small, whereas the distances between different districts may be much larger. After the discrete-model clustering, these models were further conflated, and the depth of the nodes was computed by the method that was proposed in [16].

We established LOD urban-building models in the previous step. These building models near the viewpoint have a high level of detail, whereas more distant models can be rendered by coarse models. Different from the criterion of LOD division that was proposed by [16], the original and

coarse-building models were changed into computer-screen pixels in this paper. As shown in Figure 16, when the model is rendered, then the screen pixel of model is computed:

$$\begin{aligned}
 \text{mat}_{ModelView} &= Matrix_{model} \bullet Matrix_{view} \\
 d_{depth} &= -(\text{center}[0] * \text{mat}(0,2) + \text{center}[1] * \text{mat}(1,2) + \text{center}[2] * \text{mat}(2,2) + \text{mat}(3,2)) \\
 \text{ratio} &= 2.0 * d_{Near} / (d_{Top} - d_{Bottom}) \\
 \text{screenPixel} &= \text{ratio} * \text{radius} / d_{depth}
 \end{aligned} \tag{7}$$

where  $Matrix_{model}$ ,  $Matrix_{view}$  is the model and viewport matrix in the visualization pipeline, the  $center$  is center of the model,  $radius$  is the radius of sphere bounding box for model.  $Ratio$  is the parameter for camera in the rendering, the relate parameter  $d_{Near}$ ,  $d_{top}$  and  $d_{bottom}$  in the near plane can be referenced in OpenGL manual. The screen pixel, which considered both the size of the model and the position of the camera, was used for models visualization. When the camera moves, the conflated models are dynamically scheduled and rendered.

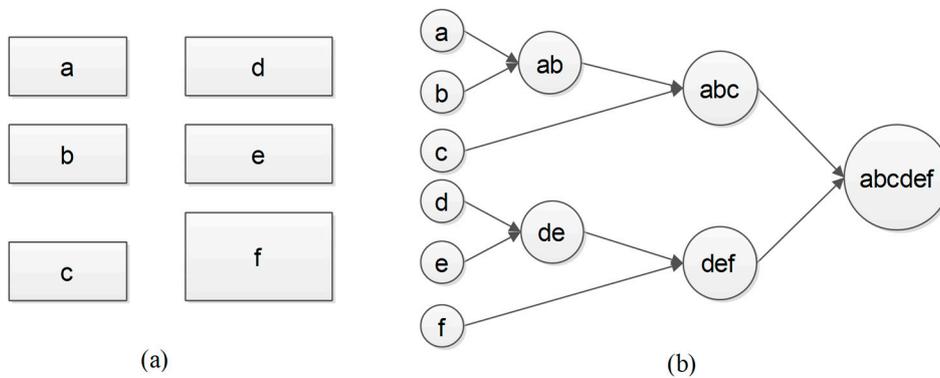


Figure 15. Model organization: (a) model’s footprints; (b) clustering order.

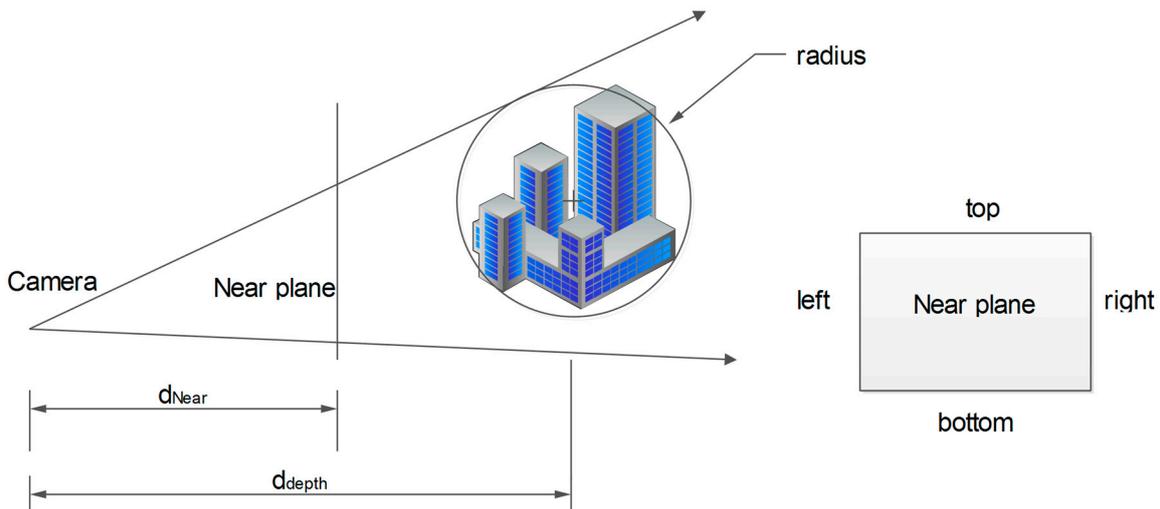


Figure 16. Level of detail (LOD) switch with pixel threshold.

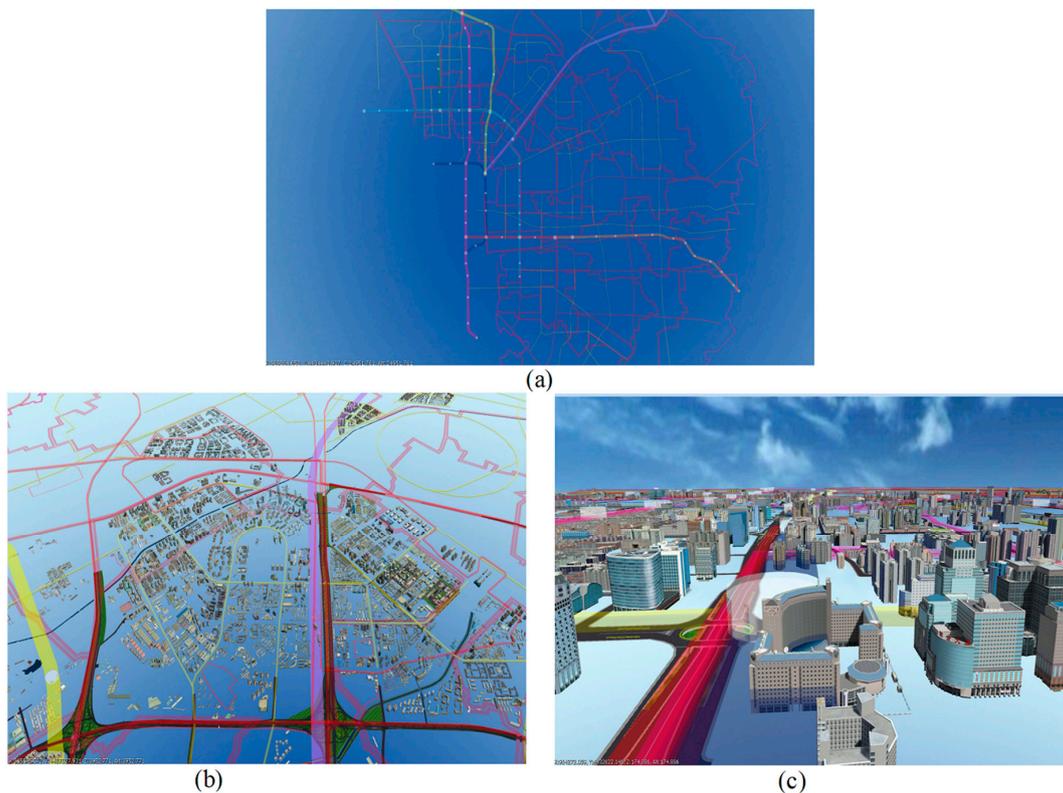
This approach considers both the distance between the viewpoint and building models and the geometrical complexity of the cluster itself. Large physical objects, such as landmarks, are usually much taller and more complex than their surrounding objects. Nodes that consist of a merged landmark and surrounding buildings are always large. Thus, landmarks can be rendered even if they are located far from the viewpoint.

#### 4. Experiment

We used an experimental platform on a Microsoft Windows 7 operating system to verify the proposed method. The clustering algorithms were realized in MATLAB 7.0, and the visualization of the 3D-model groups was based on the OpenGL development kit. The experiments were performed on a personal computer with a 1.7-GHz Intel (R) Core(R) I7 CPU, 4 GB of main memory and a NVIDIA QUADRO FX880M graphic card. Evaluating the visualization results is very difficult, and the adequacy of a generalization and visualization result depends on various factors.

For this paper, we selected a typical district in Beijing as the experimental area. The total number of buildings was 56,783; these man-made models were mainly constructed using the 3ds Max/Maya Software. The total size of the original texture image is 11.5G, and the size of geometry mesh is about 1.5 G, the total number of triangles and vertices is, respectively, 106,524,968 and 319,574,904. After conflation, the volume of texture including the original textures is 12.6 G, and the geometry mesh increase to 1.86 G by LOD generation and the total number of triangles and vertices is, respectively, up to 131,025,710 and 393,077,130. The model-clustering time was 45 min, while the model conflation lasted 1 h and 24 min because of the large amount of triangles that were sorted and removed.

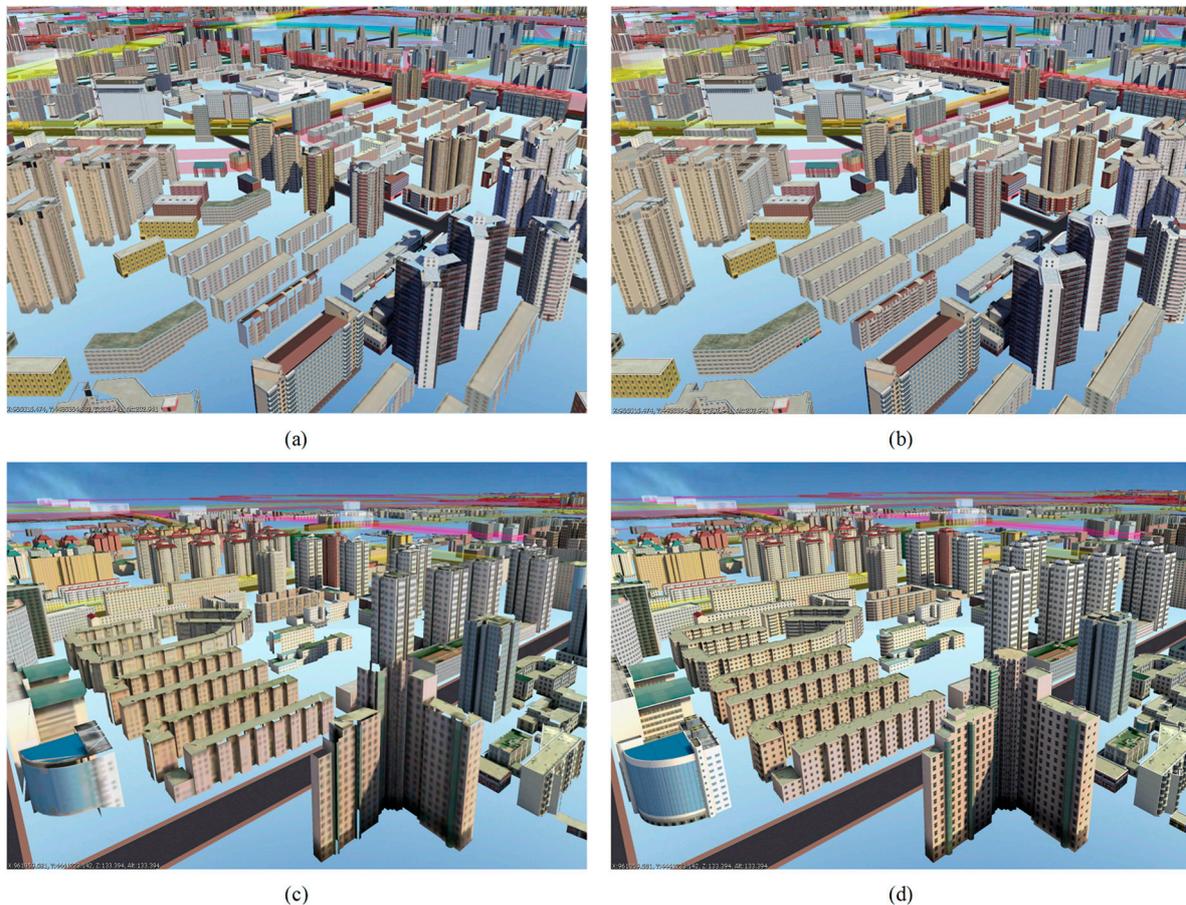
During model conflation, the multiplication coefficient  $\delta$  was 0.2. This value has been tested for these building models in the Beijing district, and it must be retested in other regions. During rendering, the pixel threshold of the LOD was 120 pixels; when the screen pixel of model was below 120 pixels, more detailed models were rendered. Figure 17a shows the visualization effect from a distant viewpoint; more detailed models were rendered from closer viewpoint in Figure 17b,c.



**Figure 17.** Model visualization from the (a) far viewpoint; (b) medium viewpoint; and (c) near viewpoint.

Figure 18 shows views of the scene. Figure 18a,c shows the rendering results of the generalization models, Figure 18b,d shows the rendering results of the original building models. Some obvious cracks exist in the conflated model because of triangle removal, but these cracks could not be visualized

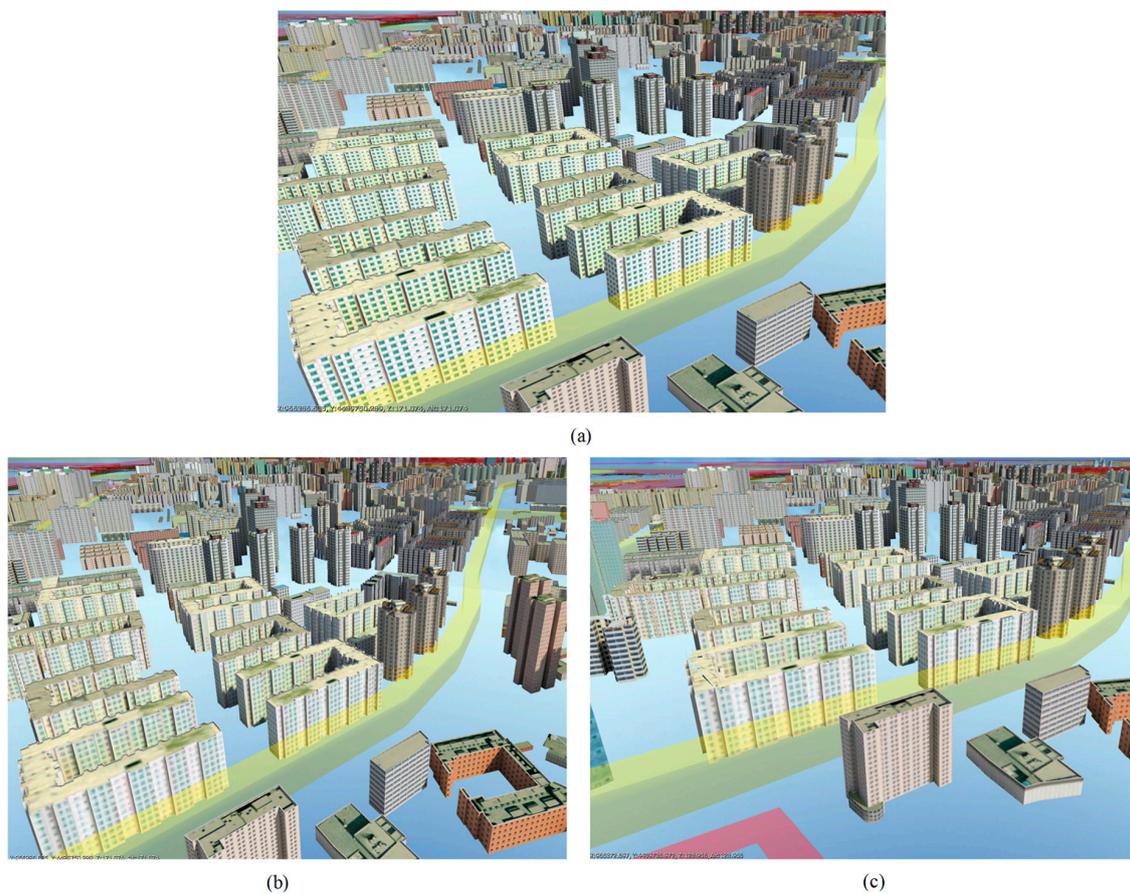
by changing the LOD threshold when rendering; thus, the original texture was preserved and the rendering speed could be improved.



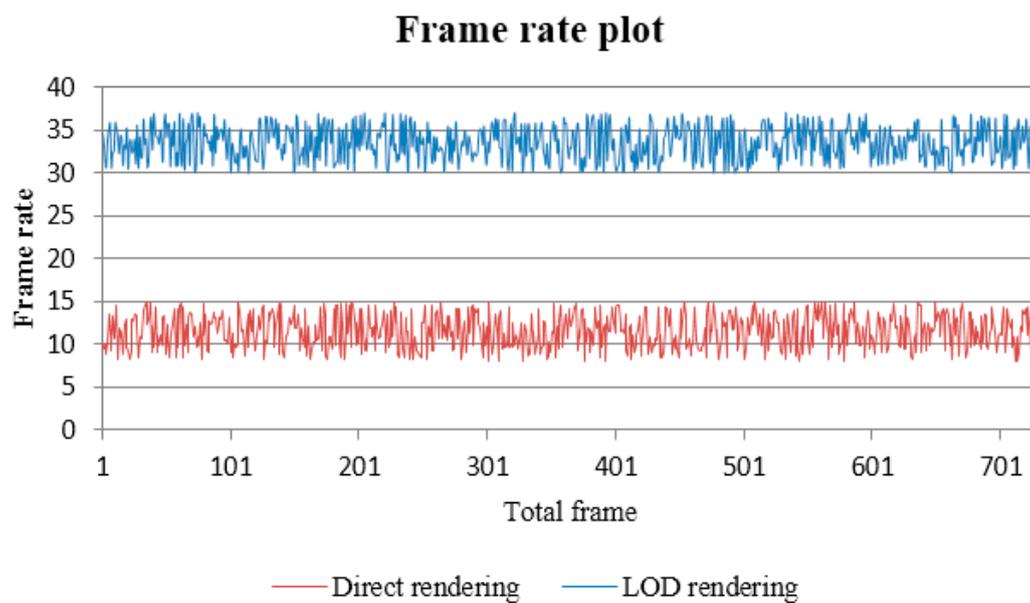
**Figure 18.** Three-dimensional building-model visualization: (a,c) conflation model with triangle removal; (b,d) original detailed-texture model.

The topological errors among the clusters barely influenced the final visual effects. Figure 19a–c shows the visualization effect of different screen-pixel thresholds when using our hierarchical method. The screen-pixel threshold was 60, 120 and 400, respectively. The different resolution group models were scheduled based on the camera position and the size of the model. When the camera moved, if the screen pixel of group models was smaller than threshold, then the corresponding more detailed models were scheduled and rendered in the scene. As shown the Figure 19, coarser models were rendered with a larger threshold, while more detailed models rendered with a smaller threshold at the same camera position.

Figure 20 illustrates the frame rates for rendering the building models with/without our hierarchical generalization method in the Beijing distinct with 56,783 buildings. These two figures demonstrate that our proposed approach effectively reduced the rendering time and struck a good balance between rendering efficiency and visual quality while maintaining the texture features. This method can notably improve the rendering speed for larger amounts of buildings during model visualization.



**Figure 19.** Building-LOD-model visualization with different screen pixels thresholds: (a) 60 pixels; (b) 120 pixels; (c) 400 pixels.



**Figure 20.** Comparison of the frame rates between rendering the origin model and LOD models.

A similar idea to our approach was proposed by Chang, Butkiewicz, Ziemkiewicz, Wartell, Ribarsky and Pollard [16]. These authors clustered buildings according to their distance and simplified

the hulls for each cluster. On the one hand, this method does not use the texture information in the clustering, and the texture can be used as an effective index for city-building clustering, which was proven in the previous section. On the other hand, the texture for each face was generated by placing an orthographic camera so that its near clipping plane lay on the face. This texture generation is very complex, and the original texture became faded; more detailed textures can be retained with our method.

Compared to existing methods [22,35], our method exhibits three significant advantages. First, the primary proximity graph is used with an MST; consequently, this method can process intricate relationships in complex urban environments. In addition, this method processes discrete polygons. Second, the texture is introduced into model generalization alongside an SOM-based classification method for texture analysis during auxiliary three-dimensional-model clustering. Landmark detection is independently introduced during generation. Third, the group models are conflated using a small triangle-removal strategy that preserves the original texture and coordinates of the image.

## 5. Discussion and Conclusions

Building-model generalization is a complex problem that has been studied by researchers for many years. This paper introduced the texture features of 3D models into the generalization process and used an SOM-based classification method for texture analysis for auxiliary three-dimensional-model clustering. The mean value and variance of the texture pixels was used to improve the classification accuracy. The results of an experimental comparison showed that this intelligent classification algorithm should be used in local regions.

In addition, this paper proposed a new hierarchical-clustering method. First, a proximity graph was constructed by CDT and visibility analysis. Each edge of the graph represented potential two-building groups, which could be used for further linear and directional detection according to the Gestalt principle. Second, texture analysis and landmark detection were independently introduced during generation. Third, models were further structurally analyzed using MST, linear detection and discrete-polygon clustering. The experimental results demonstrated the effectiveness of this algorithm.

This method is mainly proposed for a large scale of building models visualization in VGE and GIScience, a large number of applications and systems incorporate 3D city model as an integral component to serve urban planning and redevelopment, facility management, security, telecommunications, location-based services, real estate portals as well as urban-related entertainment and education products [7]. However, it does not have to be integrated for building models in small regions, building model simplification is enough. On the one hand, the model conflation is very time-consuming because of the large amount of triangles that were sorted and removed; on the other hand, the data volume is larger than the original model, which may produce transmission pressure.

This method still requires further optimization because the semantic information of 3D models has not yet been considered. Semantic identification is mainly based on names and properties, which are mainly based on similarity matching [40]. Semantic information can also identify significant landmark models. This information will be integrated into the clustering and generalization processes in future work.

**Acknowledgments:** The authors thank the editors and the anonymous reviewers for their valuable comments and suggestions, which have helped improve the context and the presentation of the article. This research is supported the National Key Technology R&D Program of China (2015BAJ06B01), and Research projects of Surveying and mapping, geographic information public welfare industry in China (201412003, 201512020), and the Fundamental Research Project of the Chinese Academy of Surveying and Mapping (7771720).

**Author Contributions:** Po Liu is mainly responsible for the writing of this paper, the main algorithm and the technical implementation. Chengming Li provides the main research idea, and relevant technical and data support. Fei Li is mainly responsible for the design and technical guidance of SOM based texture classification algorithm.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Priestnall, G.; Jarvis, C.; Burton, A.; Smith, M.J.; Mount, N.J. *Virtual Geographic Environments*; John Wiley & Sons, Ltd.: Etobicoke, ON, Canada, 2009; pp. 255–288.
2. Lin, H.; Chen, M.; Lu, G.; Zhu, Q.; Gong, J.; You, X.; Wen, Y.; Xu, B.; Hu, M. Virtual Geographic Environments (VGEs): A New Generation of Geographic Analysis Tool. *Earth-Sci. Rev.* **2013**, *126*, 74–84. [[CrossRef](#)]
3. Goodchild, M.F.; Guo, H.; Annoni, A.; Bian, L.; de Bie, K.; Campbell, F.; Craglia, M.; Ehlers, M.; van Genderen, J.; Jackson, D. Next-generation digital earth. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11088–11094. [[CrossRef](#)] [[PubMed](#)]
4. Ehlers, M. Advancing Digital Earth-Beyond the Next Generation. *Int. J. Digit. Earth* **2014**, *7*, 3–16. [[CrossRef](#)]
5. Liu, P.; Gong, J.; Yu, M. Graphics processing unit-based dynamic volume rendering for typhoons on a virtual globe. *Int. J. Digit. Earth* **2014**, *8*, 431–450. [[CrossRef](#)]
6. Liu, P.; Gong, J.; Yu, M. Visualizing and analyzing dynamic meteorological data with virtual globes: A case study of tropical cyclones. *Environ. Model. Softw.* **2015**, *64*, 80–93. [[CrossRef](#)]
7. Liang, J.; Shen, S.; Gong, J.; Liu, J.; Zhang, J. Embedding user-generated content into oblique airborne photogrammetry-based 3D city model. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1–16. [[CrossRef](#)]
8. Liang, J.; Gong, J.; Li, W.; Ibrahim, A.N. A visualization-oriented 3D method for efficient computation of urban solar radiation based on 3D–2D surface mapping. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 780–798. [[CrossRef](#)]
9. Danovaro, E.; De Floriani, L.; Magillo, P.; Puppo, E.; Sobrero, D. Level-of-detail for data analysis and exploration: A historical overview and some new perspectives. *Comput. Graph.* **2006**, *30*, 334–344. [[CrossRef](#)]
10. Fan, H.; Meng, L. A three-step approach of simplifying 3D buildings modeled by CityGML. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 1091–1107. [[CrossRef](#)]
11. Noskov, A.; Doytsher, Y. Urban Perspectives: A Raster-Based Approach to 3D Generalization of Groups of Buildings. In Proceedings of the GEOProcessing 2013, the Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services, Nice, France, 24 February–1 March 2013; pp. 67–72.
12. Zhao, J.; Zhu, Q.; Du, Z.; Feng, T.; Zhang, Y. Mathematical morphology-based generalization of complex 3D building models incorporating semantic relationships. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 95–111. [[CrossRef](#)]
13. Forberg, A. Generalization of 3D building data based on a scale-space approach. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 104–111. [[CrossRef](#)]
14. Yang, L.; Zhang, L.; Ma, J.; Xie, J.; Liu, L. Interactive visualization of multi-resolution urban building models considering spatial cognition. *Int. J. Geogr. Inf. Sci.* **2011**, *25*, 5–24. [[CrossRef](#)]
15. Karl-heinrich, A. Level of Detail Generation of 3D Building Groups by Aggregation and Typification. In Proceedings of the XXII International Cartographic Conference, A Coruña, Spain, 9–16 July 2005.
16. Chang, R.; Butkiewicz, T.; Ziemkiewicz, C.; Wartell, Z.; Ribarsky, W.; Pollard, N. Legible Simplification of Textured Urban Models. *IEEE Comput. Graph. Appl.* **2008**, *28*, 27–36. [[CrossRef](#)]
17. Lynch, K. *The Image of the City*; MIT Press: Cambridge, MA, USA, 1960; Volume 1.
18. Glander, T.; Döllner, J. Abstract representations for interactive visualization of virtual 3D city models. *Comput. Environ. Urban Syst.* **2009**, *33*, 375–387. [[CrossRef](#)]
19. Kada, M. Aggregation of 3D Buildings Using a Hybrid Data Approach. *Cartogr. Geogr. Inf. Sci.* **2011**, *38*, 153–160. [[CrossRef](#)]
20. Guercke, R.; Götzemann, T.; Brenner, C.; Sester, M. Aggregation of LoD 1 building models as an optimization problem. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 209–222. [[CrossRef](#)]
21. He, S.; Moreau, G.; Martin, J.-Y. Footprint-Based 3D Generalization of Building Groups for Virtual City Visualization. In Proceedings of the GEOProcessing 2012, The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain, 30 January–4 February 2012; pp. 177–182.
22. Zhang, L.; Hao, D.; Dong, C.; Zhen, W. A spatial cognition-based urban building clustering approach and its applications. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 721–740.
23. Garland, M.; Heckbert, P.S. Surface simplification using quadric error metrics. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 3–8 August 1997; pp. 209–216.

24. Garland, M.; Heckbert, P.S. Simplifying surfaces with color and texture using quadric error metrics. In Proceedings of the Conference on Visualization'98, Research Triangle Park, NC, USA, 18–23 October 1998; pp. 263–269.
25. Hoppe, H. New quadric metric for simplifying meshes with appearance attributes. In Proceedings of the Conference on Visualization'99: Celebrating Ten Years, San Francisco, CA, USA, 24–29 October 1999; pp. 59–66.
26. Lindstrom, P.; Turk, G. Image-driven simplification. *ACM Trans. Graph.* **2000**, *19*, 204–241. [[CrossRef](#)]
27. Patricios, N. Urban design principles of the original neighborhood concepts. *Urban Morphol.* **2002**, *6*, 21–32.
28. Regnaud, N. Contextual building typification in automated map generalization. *Algorithmica* **2001**, *30*, 312–333. [[CrossRef](#)]
29. Li, Z.; Yan, H.; Ai, T.; Chen, J. Automated building generalization based on urban morphology and Gestalt theory. *Int. J. Geogr. Inf. Sci.* **2004**, *18*, 513–534. [[CrossRef](#)]
30. Mao, B.; Ban, Y.; Harrie, L. A multiple representation data structure for dynamic visualisation of generalised 3D city models. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 198–208. [[CrossRef](#)]
31. Mao, B.; Harrie, L.; Ban, Y. Detection and typification of linear structures for dynamic visualization of 3D city models. *Comput. Environ. Urban Syst.* **2012**, *36*, 233–244. [[CrossRef](#)]
32. Zhang, L.; Han, C.; Zhang, L.; Zhang, X.; Li, J. Web-based visualization of large 3D urban building models. *Int. J. Digit. Earth* **2014**, *7*, 53–67. [[CrossRef](#)]
33. Xie, J.; Zhang, L.; Li, J.; Wang, H.; Yang, L. Automatic simplification and visualization of 3D urban building models. *Int. J. Appl. Earth Obs. Geoinf.* **2012**, *18*, 222–231. [[CrossRef](#)]
34. Cabral, M.; Lefebvre, S.; Dachsbacher, C.; Drettakis, G. Structure-Preserving Reshape for Textured Architectural Scenes. In Proceedings of the Computer Graphics Forum, Munich, Germany, 30 March–3 April 2009; pp. 469–480.
35. Zhang, M.; Zhang, L.; Takis Mathiopoulos, P.; Xie, W.; Ding, Y.; Wang, H. A geometry and texture coupled flexible generalization of urban building models. *ISPRS J. Photogramm. Remote Sens.* **2012**, *70*, 1–14. [[CrossRef](#)]
36. Glander, T.; Döllner, J. 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, 20–21 June 2008.
37. Fan, H.; Meng, L.; Jahnke, M. Generalization of 3D Buildings Modelled by CityGML. In *Advances in GIScience*; Sester, M., Bernard, L., Paelke, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 387–405.
38. Ganitseva, J.; Coors, V. Automatic landmark detection for 3D Urban models. *Archives* **2010**, *XXXVIII*, 37–43.
39. Zhang, X.; Ai, T.; Stoter, J.; Kraak, M.-J.; Molenaar, M. Building pattern recognition in topographic data: Examples on collinear and curvilinear alignments. *GeoInformatica* **2013**, *17*, 1–33. [[CrossRef](#)]
40. Al-Bakri, M.; Fairbairn, D. Assessing similarity matching for possible integration of feature classifications of geospatial data from official and informal sources. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 1437–1456. [[CrossRef](#)]

