SCIENCE CHINA
Earth Sciences

• RESEARCH PAPER •

March 2015 Vol.58 No.3: 448–461
doi: 10.1007/s11430-014-4955-8

Real-time visualization of 3D city models at street-level based on visual saliency

MAO Bo[1,2*], BAN YiFang[3] & Lars HARRIE[4]

[1] *Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, China;*
[2] *College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;*
[3] *Department of Urban Planning and Environment, KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden;*
[4] *Department of Physical Geography and Ecosystem Science, Lund University, Box118, SE-223 62 Lund, Sweden*

Received March 18, 2014; accepted July 22, 2014; published online October 17, 2014

Street-level visualization is an important application of 3D city models. Challenges to street-level visualization include the cluttering of buildings due to fine detail and visualization performance. In this paper, a novel method is proposed for street-level visualization based on visual saliency evaluation. The basic idea of the method is to preserve these salient buildings in a scene while removing those that are non-salient. The method can be divided into pre-processing procedures and real-time visualization. The first step in pre-processing is to convert 3D building models at higher Levels of Detail (LoDs) into LoD1 models with simplified ground plans. Then, a number of index viewpoints are created along the streets; these indices refer to both the position and the direction of each street site. A visual saliency value is computed for each building, with respect to the index site, based on a visual difference between the original model and the generalized model. We calculate and evaluate three methods for visual saliency: local difference, global difference and minimum projection area. The real-time visualization process begins by mapping the observer to its closest indices. The street view is then generated based on the building information stored in those indexes. A user study shows that the local visual saliency method performs better than do the global visual saliency, area and image-based methods and that the framework proposed in this paper may improve the performance of 3D visualization.

3D city models, street level visualization, index viewpoints, visual saliency

Citation: Mao B, Ban Y F, Harrie L. 2015. Real-time visualization of 3D city models at street-level based on visual saliency. Science China: Earth Sciences, 58: 448−461, doi: 10.1007/s11430-014-4955-8

3D city models are widely used in a variety of fields (Kolbe, 2008). Street-level visualization is one of its fundamental applications because many of our daily activities occur at street level, such as navigation and travel. This application is also one of the major advantages of 3D city models compared with traditional 2D maps.

One major challenge to street-level visualization is the cluttering of buildings. Another challenge is the performance of the visualization itself, which is especially important for mobile devices or Internet browsers that are increasingly being used for 3D location related applications (Rakkolainen and Vainio, 2001; Nurminen, 2008; Sun et al., 2009). These challenges call for efficient methods of street-level visualization.

In reality, the number of visible buildings is limited when viewed at street level, so only the visible ones should be loaded to generate a 3D scene of a particular viewpoint. Meanwhile, these buildings have varying levels of importance to the viewer due to their color, size or location. Therefore, we can further simplify the 3D scene by select-

*Corresponding author (email: bo.mao@njue.edu.cn) ing only those buildings that are most salient for the viewer.

The overall objective of this study is to improve the street-level visualization capabilities of 3D city models. There are three specific aims. The first is to develop evaluation algorithms to assess the visual saliency of each building at street level. We propose and test area, local and global-based methods to compute visual saliency. The second aim is to create a structure that is capable of storing the visual saliency information of each building in support of real-time visualization. The final aim is to improve the performance of real-time visualization for usage on devices with small processors, e.g., mobile devices or Internet browsers.

1 Research on street-level 3D visualization/ generalization and building saliency

1.1 Street-level visualization

The street level view of 3D city models is gaining attention from both industry and academic fields. Google launched its street view service on May 25, 2007 (Vincent, 2007). Other companies, such as Microsoft (Kopf et al., 2010) and Tencent (http://map.qq.com, accessed Dec 13, 2013), also provide a street view service for several cities. These street view services have been applied to many applications such as accessibility identification (Hara et al. 2013) and driver assistance systems (Salmen et al. 2012). However, these existing street view services are mainly based on images that are insufficient for certain applications requiring 3D view. Therefore, researchers are trying to construct and use object-based 3D city models at street level. For example, Xiao et al. (2009) proposed an image-based street city modeling method that takes street view images as input, generates the 3D points and lines of the buildings using SFM (structure-from-motion) and then maps the textures using image segmentation. Lee (2009) introduced a robust 3D reconstruction system that combines the SFM filter with bundle adjustment. Micusik and Kosecka (2009) modeled the 3D city at street level using Panoramic Sequences. Currently, the number of 3D city models in street view is increasing dramatically by implementing automatic 3D reconstruction methods. Considering the data volume of 3D city models in street view, the generalization of these models is required to improve visualization efficiency.

1.2 Generalization of 3D buildings

Generalization methods can be used to simplify 2D maps and 3D city models. Several studies have been performed that simplify single 3D buildings (Thiemann, 2002; Forberg, 2007; Kada, 2002; Sester and Brenner, 2000) and building groups (Anders, 2005; Mao et al., 2011; Guercke et al., 2011; Mao and Ban, 2013).

Viewpoint has been a consideration in the 3D generalization studies. Zhu et al. (2002) divided 3D city models into blocks and loaded the blocks around a viewpoint in dynamic visualization. They also selected different LoDs for 3D city models based on the block distance to the viewpoint. However, certain long-distance landmark buildings may be missed in this method. Parry et al. (2002) proposed a view-dependent structure for the simplification of high-resolution 3D city facades. However, few studies have concentrated on the selection of buildings for street-level visualization, which is the focus of this paper.

1.3 Visual saliency of buildings

According to Itti et al. (1998), visual saliency is defined as a measure of how visually important an object is to a viewer. Scientists have proven the effect of visual saliency on the human recognition process (Yantis, 2005; Cole et al., 2004; Thompson and Bichot, 2005). Visual saliency has been employed in many fields such as automatic target detection (Itti and Koch 2000), robotics (Frintrop et al., 2006) and video compression (Itti, 2004). In 3D related applications, saliency is used primarily for shape enhancement (Kim and Varshney, 2006), shape simplification (Menzel and Guthe, 2010), lighting (Lee et al., 2009), viewpoint selection (Mortara and Spagnuolo, 2009) and shape matching (Miao and Feng, 2010).

In geographic information science, saliency is used to detect landmarks on a map (Elias, 2003; Elias et al., 2005) and measure how visible/attractive a facade is when approaching a decision point (Winter, 2003; Raubal and Winter, 2002). In Raubal and Winter (2002), facade area, shape, color, visibility, cultural importance, intersections and boundary were measured and weighted to generate a value for the city object saliency. Elias (2003) determined building importance based on building use, size, number of immediate neighbors, orientation towards road, distance from road and height. These values were normalized to find the relative importance of each building. However, these building saliency definitions are proposed based on 2D maps. These existing methods are directly based on the city model itself, without considering the viewpoint. Therefore, the method proposed in this paper is suitable for street-level visualization when the visibility of the model is important and the saliency value is determined primarily by the user's viewpoint.

For street-level visualization situations, viewpoint and angle must be considered and visual saliency should be calculated based on the projection of 3D models. Three steps (extraction, activation and normalization/combination) are required to compute visual saliency (Harel et al., 2006). Itti et al. (1998) proposed a method based on motivated feature selection, followed by center-surround operations that highlight local gradients, which ultimately generates an overall result. Bruce and Tsotsos (2005) added "self-information" and "surprise" into the feature extraction process. Harel et al. (2006) developed a Graph-Based Visual Saliency method that creates activation maps based on certain feature channels and then normalizes them in a way that highlights conspicuity and allows their combination with other maps. This method has performed better than other existing algorithms such as that of Itti et al. (1998). Therefore, we reference this method as the existing image saliency extraction method.

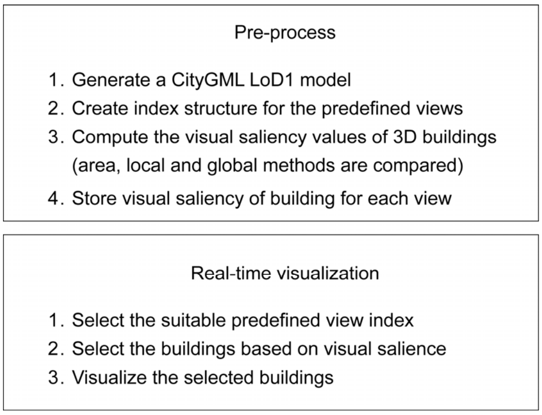
2 Street level visualization methodology

The proposed generalization method can be described simply as the selection of visually salient buildings in street view. Visual saliency describes the distinctly subjective quality that makes certain items in the world stand out from their neighbors and immediately grab our attention.

Based on the principle of visual saliency, three types of visual saliency are calculated for city models. The first is based solely on the visible area. The second is based on local difference and focuses on the change between the removed building and its view from behind. The third is based on global difference, which requires 3D city models to be represented as attributed relational graphs (ARG), the nodes of which represent the features of objects such as buildings and the edges of which represent the relationships between these objects. Then, the overall visual saliency of a building is calculated by comparing the two ARGs that represent the original model and the generalized 3D city model without that building.

2.1 Workflow of the algorithm

The workflow for the proposed street level visualization method is given in Figure 1. It contains two stages: preprocessing and real-time visualization. In the first stage, 3D city models are converted into CityGML LoD1 models that contain only the ground plan and height. Certain points along streets are then selected as viewpoint indexes. Next, for each index, the street view is generated by 3D perspective projection and the visual saliency value is estimated for each building in the index. In the second stage (real-time visualization), the current user viewpoint is taken as input and its "closest" indexes are selected. Then, the street view is generated based on stored building information from this index point and threshold values. The threshold values define the limits for building saliency values and control which buildings should be shown.


Pre-process

1. Generate a CityGML LoD1 model
2. Create index structure for the predefined views
3. Compute the visual saliency values of 3D buildings (area, local and global methods are compared)
4. Store visual saliency of building for each view

Real-time visualization

1. Select the suitable predefined view index
2. Select the buildings based on visual salience
3. Visualize the selected buildings

Figure 1 Workflow for street level visualization.

2.2 Pre-processing

2.2.1 Generating a CityGML LoD1 model

By replacing the city models with their LoD1 simplification, the process used to calculate visibility and visual saliency can be simplified and the work time can be reduced. In this study, the LoD1 representation is generated by following the guidelines outlined here (Mao et al., 2010). First, project all surfaces into the horizontal plane. Then, unify the projected surfaces to the ground plan. Finally, merge the ground plan into one polygon. In the second step, close buildings are merged and simplified. For the simplification, we use the method proposed by Sester and Brenner (2004) and extended by Fan et al. (2009).

2.2.2 Creating a predefined view index

We create the predefined viewpoints by measuring a certain distance along the roads. Each viewpoint contains two views, one for each direction in which the road can be viewed at that point. In our implementation, the road is represented as line segments. Suppose the viewpoint is on line segment (A, B); then, the two directions are AB and BA. This is illustrated in Figure 2, where the T shapes along the roads indicate the two view directions at each point.

An index structure, VPIndex, is used to store the visual saliency values of the buildings that are visible from each predefined viewpoint. The index is defined as

$$VPIndex=<key, building\_list>,$$

where *key*=<*position, orientation*> defines the view and *building_list* refers to the buildings visible from this view, sorted by their visual saliency values.

In reality, the visual saliency values vary continuously along the roads. Capturing this would require real-time computations of the visual saliency values as the user moves along the street. This would, however, require too much computation to be feasible in practice. Our pragmatic solution is to use the indexes as stated above. To ensure that visual salience values and index locations are representative, we generate the locations of the indexes as follows:

Step 1: Split the road network into proximate straight line segments.

Step 2: Create view indexes at both ends of a road segment in each direction.

Step 3: Test whether the two indexes sort visible buildings in the same order based on their visual importance.

Step 4: If not, add a new view index in the middle of the

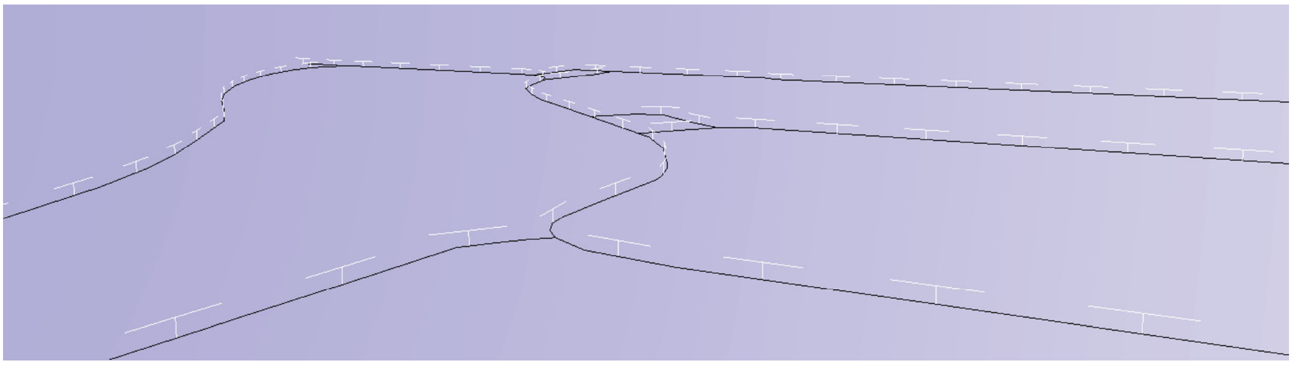


Figure 2 Generated viewpoint index along the road.

two indexes and repeat Step 3 until the neighboring indexes display the same order of visual importance for the list of buildings or until the distance between the index locations is less than a predefined threshold.

Step 5: Repeat Step 2–4 for each road segment.

2.2.3 *Calculating visual saliency components*

The objective of this study is to improve the efficiency of street-level visualization by removing buildings that are not salient. Visual saliency is introduced to measure the visual importance of each building. For 3D buildings, color, visible area and distance to center were selected as the main components of visual saliency. This section describes how these factors are computed.

(i) Color. Compared with other features, color is a difficult attribute to formalize (Anter, 2000). Therefore, the color model is simplified in this paper. In our color model, each building has only one color. The texture, lighting or viewing distance are not considered. Furthermore, if a building façade has multiple colors, then our model uses the dominant color in the textured façade as the color of the building.

From a visual saliency perspective, the color differences are more interesting than the color values. The color difference (ΔE^*_{ab}) can be defined as follows (CIE76 standard by the International Commission on Illumination, see CIE, 2007):

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2}\ , \qquad (1)$$

where ($L^*{}_1$, $a^*{}_1$, $b^*{}_1$) and ($L^*{}_2$, $a^*{}_2$, $b^*{}_2$) are two colors in Lab color space (CIE 1976 space). In our implementation, the original color values in RGB are converted into Lab color space according to Hoffmann (2003).

(ii) Visible area. The basic rule is that objects with larger visible areas should be assigned a higher saliency value. However, in 3D scenes, the visible area of a building is dependent on the viewpoint.

The visibility of each building is computed by determining whether all of its surfaces are completely covered by other surfaces. As shown in Figure 3, all of the 3D building surfaces, such as p_{i3} and p_{j3}, are projected onto a 2D visual plane (p_i and p_j). Then, the visibility of each p_i is determined in the following steps:

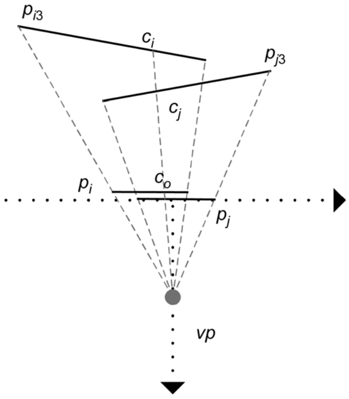


Figure 3 Polygon visibility calculation.

For every p_j ($j \neq i$), if p_j and p_i have overlapped a part of p_o, calculate c_o (the centroid of p_o).

Find two corresponding 3D points of c_o (c_i and c_j) in the original 3D polygons of p_{i3} and p_{j3}. This step is implemented by creating a line from the viewpoint *VP* to c_o and then extending the line and calculating its intersections with p_{i3} and p_{j3}.

If the distance from *VP* to c_j is smaller than that from *VP* to c_i, cut p_i with p_j.

If the area of p_i is not zero after cutting by all other p_j, define p_i as visible; otherwise, define p_i as invisible. The building is defined as visible if at least one of its polygons is visible.

(iii) Distance to center. According to the visual saliency theory, the location of an object is important. The center bias is used by Wang et al. (2010) in their image saliency computations. They note that “the closer a pixel is to the center of image, the higher probability it is observed”. In the case of street view visualization, the vanishing point of the road usually draws more attention than do other pixels in the image. In our implementation, the viewpoint is distributed along the road, so the vanishing point is usually at the center of the projection surface. Therefore, the buildings projected near the center of the image are assigned higher saliency values than are those that are projected farther from the center.

Suppose that the center of the projection plane is (0, 0) and the centroid of a building is projected into (x_c, y_c). In this case, the distance from the centroid to the projection center is $d_c = \sqrt{x_c^2 + y_c^2}$. Assume that the visual saliency value of a building is vs_b; the new adjusted visual saliency, based on distance to center, is represented by $vs_b'= vs_b/(1+d_c \times w_{dc})$, where w_{dc} is the weight of d_c. The weight is determined based on the projection parameters and the size of screen.

2.3 Computation of visual saliency values for 3D buildings

In this paper, the buildings are removed to simplify the models. Therefore, the visual saliency of a building is calculated based on the difference between the original model and the generalized model in which the building has been removed. As discussed in section 3.2.3, visual saliency is primarily defined by three factors: color, visible area and distance to center. These factors are summarized to compute the visual saliency values of a 3D building.

Three methods are proposed to calculate the saliency value: minimum projection area, local difference and global difference. The minimum projection area method only considers visible area in the saliency calculation. The local difference method considers visible area, color difference from the rear view of the building and the distance to center, but not the relationship of a building to other buildings. Finally, the global difference method calculates the overall difference between the original and the generalized models. The parameters of color and visible area were selected mainly based on the visual saliency components defined by Itti et al. (1998). Meanwhile, in street-level visualization, there is usually a road that draws the user's view into the center of the scene, so it is necessary to give more weight to the buildings near the center. Other features, such as the shape of a building, can be represented by the projection area and color difference because, in our study, we are trying to identify non-salient buildings for which visual saliency values can be defined based on the selected parameters.

2.3.1 Minimum projection area method

In street-level visualization, buildings with relatively larger visible areas usually draw more attention than do others. Therefore, the visual saliency of a building can be determined by its visible area in the projected 2D plane.

In the minimum projection area method, if the projected 2D visible area of building b_r is a_r, then $Saliency(b_r)=a_r$, where *Saliency*() is a function to calculate the saliency value of a building. The area method is actually used as a reference point to which one can compare other methods. If several buildings are removed, then the overall visual saliency is simply the sum of each building's saliency. Because visible area is the sole factor used to calculate the saliency value in this method, there is no overlap between the removed buildings and the saliency value is not affected by the order of calculation.

2.3.2 Local difference method

In the projected 2D plane, if a building is removed, the view from its rear side (or its background) appears in its place. The background view might include other buildings, sky, ground or road. The overall difference between an image taken prior to the removal of a building and an image of its background can be used to determine a building's visual saliency.

To calculate the saliency value of building b_r, the building b_r is first removed from the original city models and the remaining buildings are then re-projected. The color differences between the visible polygons of b_r and the polygons visible from the rear side view of b_r in the new model are multiplied by their intersection area and summarized together. For buildings with different colors, the color difference between each pixel pair from b_r and b_k is summarized and divided by the number of pixels. If b_r and b_k are homochromous or if their area is smaller than a certain value, then the color difference is based on their main color. Finally, the summarized color difference is adjusted by the distance from the projected centroid of b_r to the projection center. This process is described in the following pseudo code:

```
$b_r=\{p_i | 1 \leqslant i \leqslant n_r\}$;
$Remain=\{b_k | 1 \leqslant k \leqslant n_k \cap k \neq r\}$;
$Diff$=0;
$Area_r=\sum_{i=1}^{n_r} \text{Area}(p_i)$ ;
Reproject $Remain$;
For each $p_i$ in $b_r$
{For each $b_k$ in $Remain$
  {For each $p_j$ in $b_k$
      {$P_{ij}$=Intersection of $p_i$ and $p_j$;
         If Area($p_{ij}$) is not zero
         {If $p_{ij}$ in $b_r$ and $b_k$ are homochromy
    {Color difference $cd=\Delta E^*_{ab}$( $b_r$.color, $b_k$.color); }
Else
          {Color difference $cd$=; }
          $Diff$=Area($p_{ij}$)×$cd$+$Diff$;
         $Area_r=Area_r-$Area($p_{ij}$);
      }
    }
  }
}
$Diff=Area_r \times \text{Max}(\Delta E^*_{ab})+Diff$;
$dc$=distance from centroid of $b_r$ to the camera center (0,0);
```

$Saliency(b_r)=Diff/(1+dc\times w_{dc})$;

Initially, p_i is the visible polygon of b_r and *Remain* is the remaining building models without b_r. *Diff* is the summarized color difference initialized at 0. $Area_r$ is the visible area of b_r, which becomes the background (sky or road) and is initialized at the sum area of the visible polygons of b_r.

The set *Remain* is re-projected to obtain the generalized 2D visible polygons of the remaining buildings. Then, for each visible polygon p_i in b_r, intersect p_i with every reprojected visible polygon of the remaining buildings and summarize the values of their intersection areas by their color differences into *Diff* and $Area_r$, the latter of which is reduced by their intersection areas. After the loop, $Area_r$ is multiplied by the maximum difference color and added to *Diff* because it is quite obvious if a building, or a part thereof, has completely disappeared. Finally, the distance from the projected centroid of b_r to the camera center is calculated and weighted by w_{dc}. In this paper, w_{dc} is set as $1/\sqrt{2}$ because the maximum distance from a corner of the visible projection area to its center is $\sqrt{2}$. The visual saliency of b_r is generated from *Diff* and *dc*.

When multiple buildings are removed, we can sum their local differences to establish the overall visual saliency. The *Remain* set should be the original set without the removed buildings. Similar to the area method, the order of building saliency calculation does not affect the overall value.

2.3.3 Global difference method

The third method for calculating visual saliency is based on the global difference, where the overall difference between a simplified model and an original model is used to represent the saliency value of a deleted building. Harel et al. (2006) proposed a method called Graph-Based Visual Saliency (GBVS) to calculate the saliency distribution of an image. GBVS achieved 98% of the receiver operating characteristic (ROC) area of a human-based control, compared with the 84% that was achieved using the classical algorithms of Itti et al. (1998). Their method indicates that the graph-based method can be used for saliency calculation. This method is proposed for images. In this paper, we propose a new graph-based method to calculate the global difference for 3D city models while considering the projection feature in a certain viewpoint. Mathematically, features in 3D city models can be represented by attributed relational graphs (ARG), where the nodes contain features of city objects such as buildings and the edges contain information about the relationship between nodes, such as the distance between buildings. Therefore, the problem of visual similarity between city models, as in street view, is simply converted to the problem of matching two ARGs.

A great deal of effort has been devoted to developing an efficient ARG matching algorithm. Kim et al. (2004, 2010) proposed a method using the nested structure of earth mover's distance (NEMD) to calculate the difference between two ARGs. This paper employs the NEMD method to measure the visual similarity between the original and the generalized city models.

Assume two ARGs ($\mathbf{G}$ and $\mathbf{G}'$) defined as

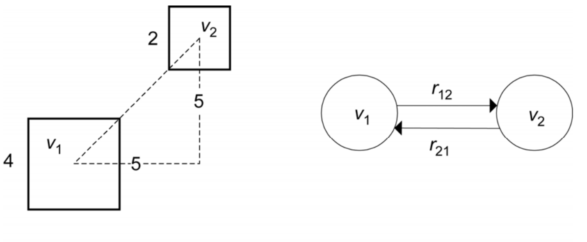
$\mathbf{G}=\{\mathbf{V},\ \mathbf{R}\}$, where $\mathbf{V}=\{v_{i,},w_i|1\leqslant i\leqslant\leq\}$ and $\mathbf{R}=\{r_{ij}|1\leqslant i\leqslant n,\ 1\leqslant j\leqslant n\}$

$\mathbf{G}'=\{\mathbf{V}',\ \mathbf{R}'\}$, where $\mathbf{V}'=\{v'_{i',},w'_{i'}|1\leqslant i'\leqslant n'\}$ and $\mathbf{R}'=\{r'_{i'j'}|1\leqslant i'\leqslant n',\ 1\leqslant j'\leqslant n'\}$

In the context of this paper, G represents an original 3D city model and $\mathbf{G}'$ represents a generalized model. All v_i and $v'_{i'}$ are v-dimensional vectors to represent features of the buildings; w_i and $w'_{i'}$ are their respective weights. All r_{ij} and $r'_{i'j'}$ are r-dimensional vectors for relationships between buildings.

Because the correspondence from $\mathbf{V}$ to $\mathbf{V}'$ is unknown, we calculate the difference from $\mathbf{V}$ to $\mathbf{V}'$ under the assumption that v_i in $\mathbf{V}$ is generalized into $v'_{i'}$ in $\mathbf{V}'$ and create a distance matrix for all of $\{v_i,\ v'_{i'}|1\leqslant i\leqslant n,\ 1\leqslant i'\leqslant n'\}$. This distance matrix provides the corresponding relationships between buildings from original and generalized models. If the minimum value in the *i*th row is located in the *i*′-th column, the v_i in $\mathbf{V}$ is converted into $v'_{i'}$ in $\mathbf{V}'$. The saliency value between $\mathbf{V}$ and $\mathbf{V}'$ is defined as the sum of the distances between their corresponding building pairs. The saliency computation is a three-step procedure. First, define the nodes and relations, as well as the distances between nodes and between relations. Second, supposing that building v_i in the original city model is generalized into $v'_{i'}$, compute the difference between $\mathbf{V}$ and $\mathbf{V}'$ with earth mover's distance (EMD) and create the $n\times n'$ distance matrix $\mathbf{D}_{\text{building}}$ for every v_i and $v'_{i'}$. Third, determine the mapping from building v_i to $v'_{i'}$ that minimizes the overall difference based on the distance matrix $\mathbf{D}_{\text{building}}$ and define the minimized difference as the saliency value between $\mathbf{V}$ and $\mathbf{V}'$.

Step 1: Definition. In our application, to reflect the visual saliency, node v_i contains two features from the building: visible projected area and color. The weight of each node is its visible projected area divided by the maximum area in the models. The relationship between two nodes is defined as the difference between the centroids of their projected polygons (dx, dy) in street view. The difference associated to the graph edge is actually a 2D vector that indicates not only the length of the difference but also the direction of the difference, or the difference in 2D space. Figure 4 gives an example of two buildings (a) and the Then, we need to define the distance between the nodes and the distance between the relationships. To make the results comparable, the distances are normalized to [0, 1], where 0 represents two items that are completely the same and 1 represents two items that are completely different. The distance between two nodes is defined by both the normalized area difference and the color difference. As shown in eq. (2), the color difference ΔE^*_{ab} is given by eq. (1) and is normalized to [0, 1] by dividing it by the maximum possible



(a) Projected buildings (b) ARG

Figure 4 Buildings in street-level view and the corresponding ARG.

color difference. The weights of area difference and color difference are the same in this paper (0.5) and could be changed according to the application requirements. Hence, we obtain the following expression (d_{node}) for the relationship between two buildings (v_i and v_j):

$$d_{\text{node}}(v_i, v_j) = 0.5 \times \frac{|Area_i - Area_j|}{MaxArea} + 0.5 \times \frac{\Delta E^*_{ab}}{Max\Delta E^*_{ab}}. \quad (2)$$

The relationship difference refers to the Euclidean distance between two buildings divided by the maximum distance between the nodes in the projected scene, as given in eq. (3), where $r_{ij}=(dx_{ij}, dy_{ij})$, $r'_{i'j'}=(dx'_{i'j'}, dy'_{i'j'})$.

$$d_{\text{relation}}\left(r_{ij}, r'_{i'j'}\right) = \sqrt{\left(dx_{ij} - dx'_{i'j'}\right)^2 + \left(dy_{ij} - dy'_{i'j'}\right)^2}. \quad (3)$$

Applying the d_{note} and d_{relation} defined in eqs. (2) and (3), we can calculate the distance between the projected 3D city models.

Step 2: Generating the difference matrix. Assuming that building v_i corresponds to building $v'_{i'}$, the difference between the original model **V** and the generalized model **V**′ is calculated using the EMD algorithm. The two models are described as $\mathbf{V}=\{(v_i, w_i)|\ 1\leqslant i\leqslant n\}$ and $\mathbf{V}'=\{(v'_{i'}, w'_{i'})|\ 1\leqslant i'\leqslant n'\}$, where v_i and v'_i are the buildings and w_i and $w'_{i'}$ are their respective weights. When building v_i corresponds to building $v'_{i'}$, the distance between v_j in V and $v'_{j'}$ in **V**′, $d_{\text{inner}}(j, j')$ is defined as the weighted sum of their feature difference ($d_{\text{node}}(v_j, v'_{j'})$) and their relation difference from v_i and $v'_{i'}$, respectively ($d_{\text{relation}}(r_{ij}, r'_{i'j'})$), as shown in eq. (4):

$$d_{\text{inner}}\left(j, j'\right) = \left(1-p\right) \times d_{\text{node}}\left(v_j, v'_{j'}\right) + p \times d_{\text{relation}}\left(r_{ij}, r'_{i'j'}\right), \quad (4)$$

where p is in the interval [0, 1] that adjusts the weights of nodes and relations. In our application, p is set to 0.5, which means that the node and relation are equally important. Based on eq. (4), we obtain the distance matrix $D_{\text{inner}}=[d_{\text{inner}}(j, j')|\ 1\leqslant j\leqslant n, 1\leqslant j'\leqslant n']$.

The EMD distance from V to V' is computed by using the distance matrix D_{inner}. To obtain the EMD, the mapping matrix $F=[f(j, j')]$, where $f(j, j')$ represents the correspondence between building v_j and $v'_{j'}$, is computed by minimizing the distance given by eq. (5) (Kim et al. 2004):

$$\sum_{j=1}^{n}\sum_{j'=1}^{n'} d_{\text{inner}}\left(v_j, v'_{j'}\right) f\left(j, j'\right). \quad (5)$$

It is then subjected to the following constraints:

$$f(j, j') \geqslant 0,\ 1 \leqslant j \leqslant n,\ 1 \leqslant j' \leqslant n',$$

$$\sum_{j'=1}^{n'} f(j, j') \leqslant w_i,\ 1 \leqslant j \leqslant n,$$

$$\sum_{j=1}^{n} f(j, j') \leqslant w'_{j'},\ 1 \leqslant j' \leqslant n',$$

$$\sum_{j=1}^{n}\sum_{j'=1}^{n'} f(j, j') = \min\left(\sum_{j=1}^{n} w_j, \sum_{j'=1}^{n'} w'_{j'}\right).$$

The first constraint allows the conversion from an original model to a generalized model, not vice versa. The next two constraints limit the amount of “conversion” for each building by its weight. The fourth constraint forces the equation to convert the maximum number of buildings possible. Based on the optimal mapping **F**, the EMD is defined as the work normalized by the total weights:

$$\text{EMD}(\mathbf{V}, \mathbf{V}') = \frac{\sum_{j=1}^{n}\sum_{j'=1}^{n'} d(v_j, v'_{j'}) f\left(j, j'\right)}{\sum_{j=1}^{n}\sum_{j'=1}^{n'} f\left(j, j'\right)}. \quad (6)$$

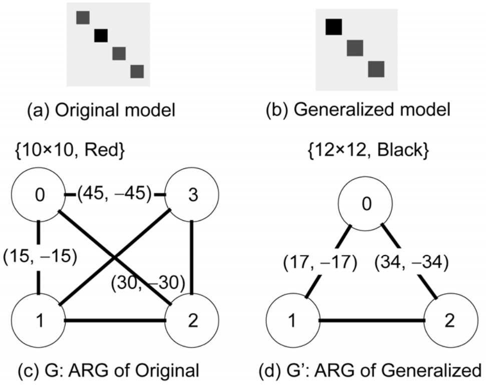
The computation of the mapping matrix is a well-known transportation problem (Hitchcock, 1941). We employ the source code supplied by Till Schulte-Coerne (2005) to implement the calculation. Eq. (6) is the inner EMD, which yields $d_{\text{building}}(i, i')$, the distance between **V** and **V**′ if v_i is converted to $v'_{i'}$. It is one element in the overall Distance matrix $D_{\text{building}}=[d_{\text{building}}(i, i')|\ 1\leqslant i\leqslant n, 1\leqslant i'\leqslant n']$.

Step 3: Calculating global saliency. The difference in the global visual saliency values VS between the original and generalized models is defined as the sum of the distances between their corresponding nodes.The distance matrix, $D_{\text{building}}=[d_{\text{building}}(i, i')|\ 1\leqslant i\leqslant n, 1\leqslant i'\leqslant n']$, is generated from the $n\times n'$ EMDs and indicates the corresponding relationship from nodes in **V** to **V**′. To reflect the overall difference, eq. (7) gives the calculation of VS for an original 3D city model **V** and a generalized model **V**′.

$$VS = \sum_{i=1}^{n} \min\{d_{\text{building}}(v_i, v'_{i'}) \times Area(v_i)\,|\, 1 \leqslant i' \leqslant n\}. \quad (7)$$

Simple example of a visual saliency computation. We provide a simplified example of how NEMD can be used to compare two views of a 3D city model. In Figure 5, (a) is the original projected city model and (b) is the generalized model created by removing one node/building and extending the remains. Items (c) and (d) in the model depict **G** and **G**′, respectively, which are the ARGs of the models. The node 0 in **G** represents the top left building in (a); its area is 10×10 and its color is red. The node 0 in **G**′ is the top left building in (b); its area is 12×12 and its color is black.

To obtain the NEMD value of **G** and **G**′, we first need to



(a) Original model (b) Generalized model

(c) G: ARG of Original (d) G′: ARG of Generalized

Figure 5 Example of NEMD calculation.

create the inner distance matrix $D_{\rm inner}$ of every node pair. For two ARGs in Figure 5, 12 $\mathbf{D}_{\rm inner}$ matrixes are created based on eq. (2). For example, $\mathbf{D}_{\rm inner}$ of v_0 and v'_0 is given in eq. (8). Then, based on the $\mathbf{D}_{\rm inner}$ that was just created, we compute one element in $\mathbf{D}_{\rm outer}$ matrix at a time by summarizing the smallest values in different columns and rows and dividing by the number of rows. The first element is 0.13, so no elements in the third column and third row are considered in the next selection round. Therefore, 0.19 and 0.20 are selected. For example, $d_{\rm outer}(v_0, v'_0)$=(0.2+0.19+0.13)/(1+1+1)=0.173; because the area is the same, the weights are 1. Including all elements, it is generated as shown in eq. (9).

$d_{\rm note}(v_0,v'_0)$=0.5×|area$_0$-area$_0$′|/MaxArea+0.5×ΔE((255,0,0), (0,0,0))/Δ$E_{\rm max}$=0.5×|100−144|/144+0.5×125.4/130=0.635,

$d_{\rm relation}(r_{00}, r'_{00})$=0 because r_{00}=r'_{00}=(0,0), $d_{\rm inner}(v_0, v'_0, v_0, v'_0)$=(1−$p$)×$d_{\rm note}(v_0,v'_0)$+$p$×$d_{\rm relation}(r_{00}, r'_{00})$=0.667×0.635+0.333×0=0.42,

$$\mathbf{D}_{\rm inner(0,0)}=\begin{bmatrix} d_{\rm inner}(v_0,v'_0,v_0,v'_0) & d_{\rm inner}(v_0,v'_0,v_0,v'_1) & d_{\rm inner}(v_0,v'_0,v_0,v'_2) \\ d_{\rm inner}(v_0,v'_0,v_1,v'_0) & d_{\rm inner}(v_0,v'_0,v_1,v'_1) & d_{\rm inner}(v_0,v'_0,v_1,v'_2) \\ d_{\rm inner}(v_0,v'_0,v_2,v'_0) & d_{\rm inner}(v_0,v'_0,v_2,v'_1) & d_{\rm inner}(v_0,v'_0,v_2,v'_2) \\ d_{\rm inner}(v_0,v'_0,v_3,v'_0) & d_{\rm inner}(v_0,v'_0,v_3,v'_1) & d_{\rm inner}(v_0,v'_0,v_3,v'_2) \end{bmatrix}=\begin{bmatrix} 0.42 & 0.20 & 0.31 \\ 0.19 & 0.44 & 0.54 \\ 0.60 & 0.18 & 0.13 \\ 0.69 & 0.27 & 0.17 \end{bmatrix}. \quad (8)$$

$$\mathbf{D}_{\rm outer}=\begin{bmatrix} d_{\rm outer}(v_0,v'_0) & d_{\rm outer}(v_0,v'_1) & d_{\rm outer}(v_0,v'_2) \\ d_{\rm outer}(v_1,v'_0) & d_{\rm outer}(v_1,v'_1) & d_{\rm outer}(v_1,v'_2) \\ d_{\rm outer}(v_2,v'_0) & d_{\rm outer}(v_2,v'_1) & d_{\rm outer}(v_2,v'_2) \\ d_{\rm outer}(v_3,v'_0) & d_{\rm outer}(v_3,v'_1) & d_{\rm outer}(v_2,v'_2) \end{bmatrix}=\begin{bmatrix} 0.173 & 0.192 & 0.293 \\ 0.113 & 0.169 & 0.204 \\ 0.204 & 0.109 & 0.165 \\ 0.293 & 0.192 & 0.113 \end{bmatrix}. \quad (9)$$

$\mathbf{D}_{\rm outer}$ shows the distance between nodes from two ARGs and this distance indicates the corresponding relation between nodes. For example, in eq. (9), the minimum value in the second row is $\mathbf{D}_{\rm outer}(v_1, v'_0)$, which means that node 1 in G is generalized into node 0 in G′. The minimum value in the first row of eq. (9) is $\mathbf{D}_{\rm outer}(v_0, v'_0)$, which indicates that the node v_0 is merged into v'_0 in the generalization process. The NEMD value of the two ARGs is generated from the $\mathbf{D}_{\rm outer}$ by summarizing the smallest values in each row to reflect the influence of removing nodes (the row number is the original node number and the column number is the selected node). Therefore, the NEMD value between **G** and **G**′ is 0.173+0.113+0.109+0.113=0.508.

2.4 Storing visual saliency values of 3D buildings

As described in section 3.2.2, the viewpoint index is defined as *VPIndex=<key*, *building_list>*. When the visible building list of an index is generated, the visual saliency values are calculated for each of the buildings in the list. The saliency value of a building is calculated to be the visual distance between the original model and the model without the building (for any of the three methods proposed above). In pseudo code, the process is as follows:

```
Original={All visible buildings};
Remain={All visible buildings};
Result={};
While(Remain!=NULL){
   For each building $b_i$ in Remain{
      $d_i$=Saliency(Remain−$b_i$, Original);
   }
   Select the $b_{\rm min}$ with minimum $d_i$ ($d_{\rm min}$);
   Set the visual saliency value of $b_{\rm min}$ into $d_{\rm min}$;
   Remove $b_{\rm min}$ from Remain;
   Insert $b_{\rm min}$ into the front of Result.}
```

In the pseudo code, all visible buildings are saved in a set called *Original*. In the beginning, the set *Remain* contains all visible buildings and the set *Result* is empty. Then, for each building b_i in *Remain*, the difference in saliency values between the *Remain* set without b_i and the *Original* set is calculated. The calculation function, *Saliency*(), can be the area difference, local difference, or global difference. Next, select the b_i with the minimum d_i as $b_{\rm min}$, remove $b_{\rm min}$ from set *Remain* and insert it into the front of the set *Result*. By storing each minimum d_i value, we can easily obtain the visual difference between the generalized models and the original ones.

2.5 Real-time visualization

The first step in the real-time visualization process is to map the current user's view to a predefined view (stored in the *Viewpoint index*). Assuming that the user viewpoint (*vp*) is restricted along the road, this mapping is done by selecting the two nearest predefined viewpoints that are set in the same direction as the user's view. As shown in Figure 6, r_1 and r_2 are the selected viewpoints for the current user's view *vp*. Because the predefined views are different from the current user's view, there may be some missing buildings if we replace the visible buildings at *vp* by r_1 or r_2. To account for as many visible buildings in *vp* as possible, r_1 is a more reasonable choice than r_2, though r_2 may be closer to *vp* than r_1. The detailed coverage analysis is given in this section.

The visible buildings in the selected predefined view are generalized by their visual saliency values and visualized for the current user. The generalization strategy should be adjusted for different applications by resetting the thresholds that determine which buildings are visualized.

When the user moves around the models at street level, we continuously check the user's viewpoint and find its corresponding predefined view, which is indicated by current indexes CI. If the new selected indexes are the same as the current indexes CI, then nothing is done; otherwise, CI is set as the new indexes, those visible buildings are loaded and the 3D scenes are refreshed accordingly.

Although the user viewpoint is located nearby the predefined viewpoint index, there is still a certain distance from the index. The distance d_{vp} is in [0, *step*], where *step* represents the interval between neighbor indexes, as shown in Figure 6. Therefore, some buildings that are visible in *vp* may not be visible in r_1. To reduce the number of discrepancies, two methods are tested. One is to reduce the *step* and another is to include both of the indexes that are near the viewpoints (r_1 and r_2, as shown in Figure 6). We calculate the number of missing buildings using different index selection strategies by randomly generating the viewpoint along the street. Table 1 gives the distribution of missing building numbers for each index selection strategy. In Table 1, *Step*=*N* means that the *step* is set to *N* meters and *Step*=*N*(2) means that the step is set to *N* meters and the visible models in both nearby indexes are selected.

From Table 1, we can see that the visualization accuracy is increased by using two nearby indexes and that there is less improvement when only the strategy for reducing the intervals between indexes is used. Meanwhile, the average number of visible buildings in two neighbor indexes is not as high as the sum of visible buildings in the two indexes because many buildings are visible in both indexes. Table 2 gives the average visible number in different steps.

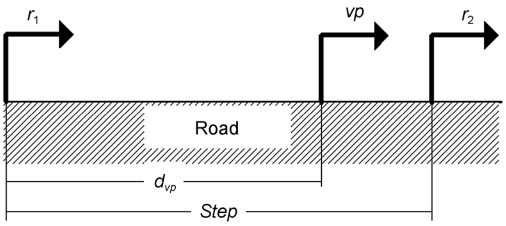


Figure 6 Viewpoint-based index selection.

Table 1 Distribution of missing buildings in different index strategies

Selection strategy	Distribution of the No. of missing buildings			
	0	1	2	>2
Step=40	62%	18%	10%	10%
Step=30	68%	17%	6%	9%
Step=20	72%	16%	6%	6%
Step=10	81%	13%	4%	2%
Step=5	88%	9%	2%	1%
Step=40(2)	79%	17%	3%	1%
Step=20(2)	92%	7%	1%	
Step=10(2)	95%	4%	1%	

Two indexes are involved in the visualization of a viewpoint, so some buildings may overlap and have two visual saliency values. The selection is based on the maximum visual saliency, which preserves important buildings, though some invisible buildings may also be preserved.

3 Case study

3.1 Implementation

The experimental environment was implemented in Java. The platform was Eclipse 3.4.1 running on a PC with Inter 2.4 GHz Core2 Duo CPU, 2.39 GHz 3.25GB RAM and Microsoft Windows XP SP3. The CityGML data were parsed by citygml4j 0.2.0 (CityGML4j, 2013, http://opportunity.bv.tu-erlin.de/software/projects/show/citygml4j,accessed Nov 12, 2013). The 3D city model was visualized with Xj3D 2.0.0 (Xj3D, 2013). The test datasets were obtained from CityGML.org (with random generated textures).

The visualization framework is given in Figure 7. CityGML datasets were parsed with CityGML4j and converted into Java objects representing City Objects, such as buildings and roads. These were then converted to city object classes and assigned to one or several X3D Scenes with JTS (2013) and Xj3D (2013). The Java X3D scenes were viewed in Xj3D Viewer (Xj3D, 2013) and the user interaction information (e.g., viewpoint location, orientation) was obtained from the viewer. Based on the viewpoint index information, the generalized scenes are generated dynamically.

Table 2 Average node numbers in different index selection methods

	Step=40	*Step*=20	*Step*=10
One index	8.662	8.853	8.711
Two neighbor indexes	10.518	9.75	9.277

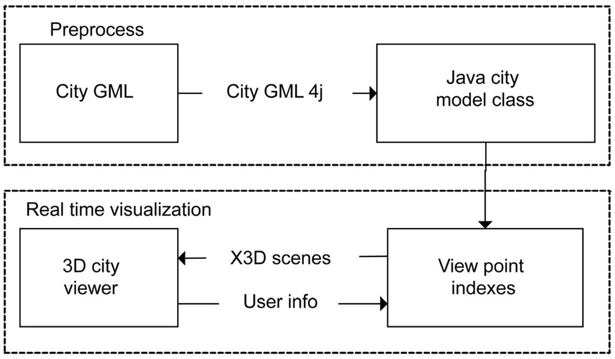


Figure 7 Framework of the 3D city model preprocess and visualization.

3.2 User study of street level visualization

(i) Method. To evaluate the three proposed visual saliency methods, a user survey was conducted. The user survey included five questions about how similar the generalized street views were to the original view. In each question, three generalizations were created using the proposed local difference, global difference or minimum projection area method. In questions 1–5 (Q1–Q5), the reduced building numbers were 2, 3, 4, 5 and 6. An example of the question is given in Figure 8, in which Images (b)–(d) were generated by removing 5 buildings according to (b) their minimum projection area, (c) minimizing the local difference and (d) minimizing the global difference. The different areas from the original model were highlighted with a red circle, as shown in Figure 8. The user was asked to order these three images (without the highlights) according to their similarity to the original image.

The user test was conducted by 38 specialists in the field (colleagues at KTH and Lund University). They received the user queries by email and viewed the images on the screen.

(ii) Result. Table 3 provides the result of the user study. To digitize the users' replies, the image ranked as most similar was given 2 points, followed by 1 point and 0 point. All user replies were averaged. Therefore, an image assigned 2 points would imply that everyone thought it was the most similar to the original image. The average values of the user replies and the local visual saliency are listed in Table 3.

(iii) Discussion. From Table 3, we can conclude that the majority of case study participants replied that minimizing the local difference created the most similar image to the original image. When the number of removed buildings is small (2–4 in Q1–Q3), we can assume that it is difficult for users to choose between the area projection method and the global difference method because these methods provided quite similar results. However, when more buildings are removed (Q4 and Q5), users replied that the global method provided better results than did the area method.

In Table 4, the saliency values computed using the local difference method (denoted local saliency) for all images are given. By comparing Tables 3 and 4, we can identify a strong relationship between local saliency and user preference. An increasing difference between the local saliency value (in the three images) implied a clearer vote for the local saliency difference method, as illustrated in Figure 9. The graph is constructed as follows. Let n_u be the average point of the best results (local difference method in all tests), e.g., n_u=1.55 in Q1. Then, suppose $n'_u=(n_u-1.5)\times 5$ and $d_l=(d_{\min}-d_{\min 2})\times 100$, where $d_{\min}$ is the minimum local saliency value and $d_{\min 2}$ is the second lowest local saliency value in Table 4.

During the survey, many users replied that the generalization results in Q1, Q2 and Q3 were quite similar to the original one and that it was difficult to identify the difference, which demonstrates that it may be safe to remove some visually unimportant buildings in street-level visualization.

3.3 A comparison with GBVS

We compare our methods with the image based saliency method, Graph-Based Visual Saliency (GBVS), proposed by Harel et al. (2006). According to their test, GBVS achieved 98% of the receiver operating characteristic (ROC) area of a

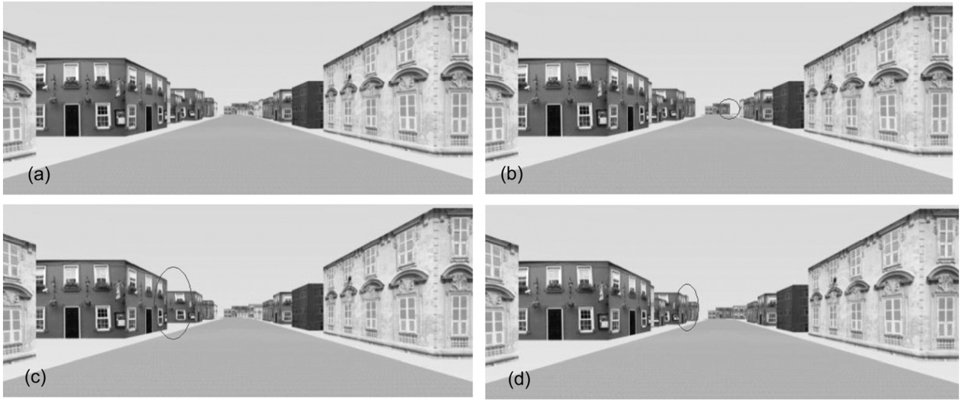


Figure 8 Question in the user test. (a) Original image; (b) minimum projection area; (c) minimize the local difference; (d) minimize the global difference.

Table 3 User survey results

	Minimum area projection	Local difference	Global difference
Q1	1.0	1.55	1.05
Q2	0.66	1.66	0.86
Q3	0.63	1.71	0.52
Q4	0.26	1.84	1.1
Q5	0.34	1.79	1.21

Table 4 Saliency values computed using the proposed method

	Minimum area projection	Local difference	Global difference
Q1	0.0107	0.00422	0.010478
Q2	0.022987	0.008342	0.020752
Q3	0.0330	0.01611	0.0330
Q4	0.05992	0.01987	0.045869
Q5	0.064044	0.030146	0.049986

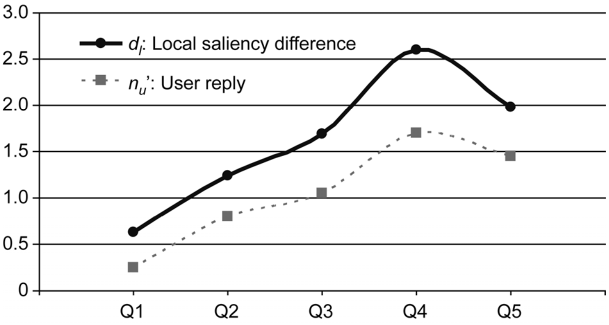


Figure 9 Correlation between user reply and local saliency difference.

human-based control, compared with the 84% obtained using the classical algorithms of Itti et al. (1998). This paper tests the ability of the GBVS method to measure the visual similarity between an original and a generalized city model. We use the implementation of GBVS found in Harel (2013). This Matlab implementation is integrated with Xj3D in the Eclipse development platform.

We first generate the image of the 3D scene with Xj3D render engine. However, the lighting of the scene may change based on time and environmental settings, such as sun light, day/night, or season, so we set parallel light as the predefined light and blue sky as the background.

Assume that Img represents the projection of an original 3D city model and Img' represents a generalized model. We then compute the GBVS map of both Img and Img' as gbvs and gbvs'. The visual saliency difference of two images is represented as the sum of all elements of |gbvs-gbvs'|. The results of the GBVS are given in Figure 10.

The most salient areas are located near the sides of the images, as indicated in Figure 10(c) and (d). Therefore, it is difficult to identify the less salient buildings. According to our experiment, the GBVS method cannot calculate the visual saliency effectively. We performed the GBVS on the images from Q1 to Q5 and the saliency did not increase in conjunction with the number of removed buildings, which is not supported by the user survey. In fact, the image-based visual saliency algorithms are mainly focused on the visually important buildings, but they cannot address those that are less salient. The results also indicate that the consideration of too many features (GBVS has eight features including color, shape, orientation and flicker) does not improve the accuracy of saliency computation at street level.

3.4 Performance

There were 90 buildings in the test area. Two view indexes were created along the two directions of the road every 20 meters. A total of 368 indexes were generated in the test area. We repeated the process of image loading 30 times to calculate the average time. The average load time was linearly related to the number of buildings loaded; therefore, if the model were to become too large, the load time would become unacceptable. By using the index based street level visualization method, the number of loaded buildings is dramatically reduced (from 90 to 9.75). Because we can control the number of buildings being visualized by their saliency values, the number can be reduced even further. In our implementation, the viewpoint indexes were stored in a list, where the complexity of matching is O(n). For large city models, these indexes can be better organized to increase the matching speed.

Preprocessing was implemented and tested in the same platform. After 10 tests, the average time for generating the indices was approximately 103 s for 90 buildings when using the minimum color difference method. For the minimum area method, the index creation time was approximately 30 s and it took more than 10 min for the NEMD based global difference method. Because the index can be generated offline, it would be acceptable if the process were to take some time.

In the visualization step, the relationship between FPS (Frame per Second) and the number of loaded models was tested in a web access environment. The 3D city models were converted into X3D files, which were rendered directly in the Internet browser using X3DOM technology (Behr et al., 2009). This platform was implemented by Mao and Ban (2011) and the browser is Mozilla Firefox 4.06bpre. Figure 11 gives the relationship between the stable FPS in the browser and the loaded number of textured buildings and indicates that the FPS is quite low (<5) when the building number is more than 10. Therefore, it is essential to reduce the building number, especially in web/mobile 3D visualization situations. The proposed generalization method is capable of doing just that and should be applied in web/mobile related 3D city visualization applications.

4 Discussion

In this section, the results of local and global saliency

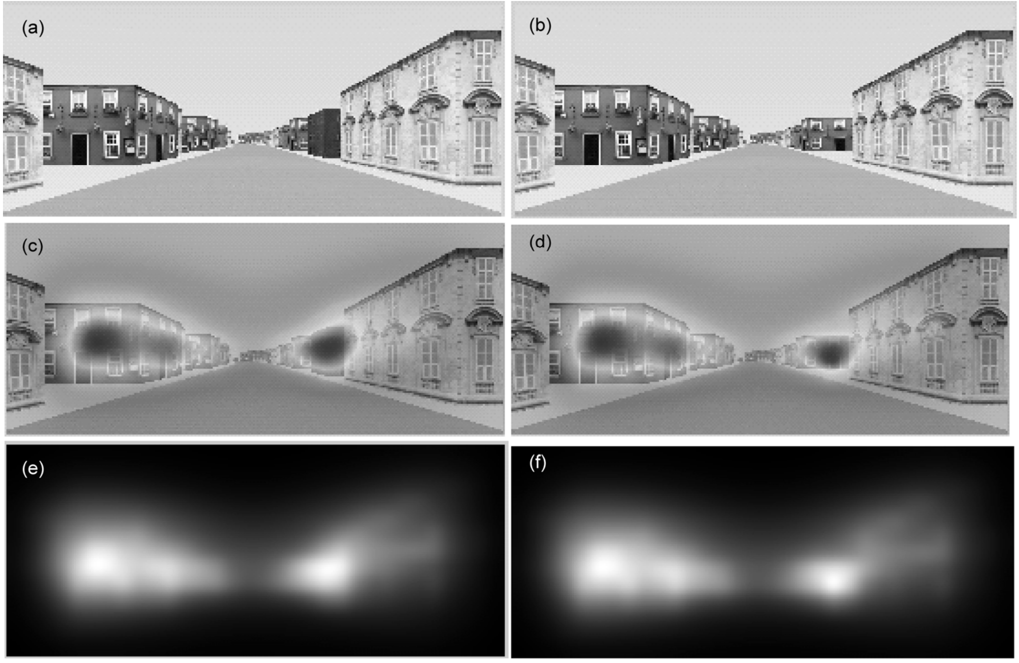


Figure 10 Visual saliency using GBVS. (a) Original image; (b) generalized (one building removed); (c) original GBVS overlap map; (d) generalized GBVS overlap map; (e) original GBVS value; (f) generalized GBVS value.

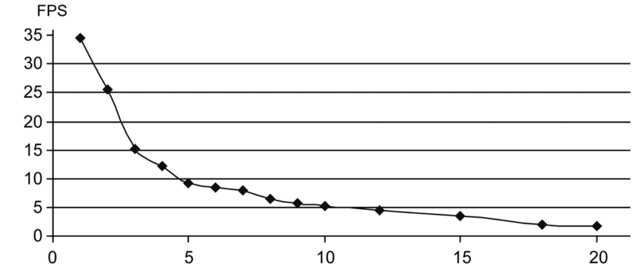


Figure 11 FPS and building number in X3DOM.

calculation methods are analyzed. The performance of the proposed dynamic visualization method is further discussed, especially in relation to mobile/web applications.

4.1 Analysis of local and global difference methods

According to the user survey, the local difference method provides better results than does the global difference method because in street-level visualization, there are usually no clear patterns in the overall level. For example, the buildings along the road are completely different from each other. Therefore, it is quite difficult for people to remember a whole picture at street level that contains so many details. The users focus on local landmarks and therefore mainly recognize differences at the local level. That is why the local difference method is more suitable than the global difference method for street-level visual saliency calculations.

Compared with image-based saliency calculation methods such as GBVS, which consider the 3D scene as an image and consider features such as shape, color, orientation and flicker, our results indicate that complex calculations with too many considerations do not substantially improve the results. The sample color difference method produces the best results according to user experience. The color difference method is more effective at street level, where most of the buildings are rectangular in shape. Therefore, the proposed method can deal well with street-level visualization situations.

4.2 Dynamic visualization in mobile/web applications

Another phenomenon that we observed based on user survey results is that most users have difficulty in identifying image differences when the computed local visual saliency value of a particular building is small (generalization results of Q1–Q3 in Table 4). This finding supports our initial hypothesis that removing some buildings with small visual saliency values does not affect the main features of 3D city models at street level. This is quite an important discovery in regard to the applications that are being implemented with limited computational resources, such as mobile/web applications. On the one hand, these applications cannot address a whole city, especially in the LoDs used at street level. As shown in section 4.3, the number of buildings loaded has a strong influence on system performance. On the other hand, it is time consuming to calculate visual saliency in real time. Therefore, the visual index structure can save the repeated computation and supply the required similarity. Additionally, by selecting the buildings that are loaded based on their visual saliency, it is possible to satisfy strict limits on the number of visible buildings while achieving maximum visual similarity to the original models.

5 Conclusion

In this paper, a visualization method for street level viewing is proposed. The visual saliency value of each building within certain viewpoints is computed and stored in a structure called a viewpoint index. By using the viewpoint indexes, we can improve visual efficiency in real time. This paper provides three main contributions. First, it introduces the minimum area projection, the local difference and the global difference methods to compute visual saliency values for buildings in street view. Second, a user test demonstrates that the local difference method is the preferred method for determining visual saliency when saliency values are used for removing buildings in street view. Third, we show that it is better to use two nearby indexes than to increase the density of indexes.

The user survey indicates that the proposed methods can calculate the visual saliency for normal buildings at street level. However, in real cities, many landmarks may not be visually salient but have to be preserved for visualization. In future studies, we need to identify these landmarks based on both visual saliency (e.g., height, body shape and roof structure) and semantic information (e.g., owner, building history and purpose).

This work was supported by the National Natural Science Foundation of China (Grant No. 41201486), the National Key Technologies R&D Program of China (Grant No. SQ2013GX07E00985), and the project of the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) in the Collaborative Innovation Center of Modern Grain Circulation and Security, Nanjing University of Finance and Economics.

Anders K. 2005. Level of detail generation of 3D building groups by aggregation and typification. Proceedings of XXII International Cartographic Conference 2005, Spain

Anter K. 2000. What Colour is the Red House? Perceived colour of painted facades. Doctoral Dissertation. Stockholm: Royal Institute of Technology

Behr J, Eschler P, Jung Y, Zöllner M. 2009. X3DOM: A DOM-based HTML5/X3D integration model. Web3D '09, Darmstadt, Germany

Bruce N, Tsotsos J. 2005. Saliency Based on information maximization. Proc Neural Inf Process Syst (NIPS), 2005

International Commission on Illumination (CIE). 2007. CIE 1976 L*a*b* colour space draft standard. http://cie.co. at/index.php?i_ca_id=485. accessed Feb 13, 2014

Cole G G, Kentridge R W, Heywood C A. 2004. Visual saliency in the change detection paradigm: The special role of object onset. J Exp Psychol-Human Percept Performance, 30: 464–477

Elias B. 2003. Extracting landmarks with data mining methods. In: Kuhn W, Worboys M, Timpf S, eds. Spatial Information Theory: Foundations of Geographic Information Science. Lecture Notes Comput Sci, 2825: 375–389

Elias B, Hampe M, Monika S. 2005. Adaptive visualisation of landmarks using an MRDB. In: Meng L Q, Zipf A, Reichenbacher T, eds. Map-based Mobile Services-Theories, Methods and Implementations. Heidelberg: Springer. 73–86

Fan H, Meng L, Jahnke M. 2009. Generalization of 3D buildings modeled by CityGML. Lecture notes in geoinformation and cartography. Adv GISci, 387–405

Forberg A. 2007. Generalization of 3D building data based on a scale-space approach. ISPRS-J Photogramm Remote Sens, 62: 104–111

Guercke R, Götzelmann T, Brenner C, et al. 2011. Aggregation of LoD 1 building models as an optimization problem. ISPRS-J Photogramm Remote Sens, 66: 209–222

Hara K, Le V, Froehlich J. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In: Proc of the SIGCHI Conference on Human Factors in Computing Syst (CHI '13). New York: ACM. 631–640

Harel J. 2013. A Saliency Implementation in MATLAB: http://www.klab.caltech.edu/~harel/share/gbvs.php. accessed Nov 12 2013

Harel J, Koch C, Perona P. 2006. Graph-based visual saliency. In: Schölkopf B, Platt J C, Hoffman T, eds. Advances, Proceeding of Neural Information Process Systems 19 (NIPS 2006)

Hitchcock F L. 1941. The distribution of a product from several sources to numerous localities. J Math Phys, 20: 224–230

Hoffmann G. 2003. CIELAB Color Space, online at: http://www.fho-emden.de/~hoffmann/cielab03022003.pdf. accessed Nov 12, 2013

Itti L. 2004. Automatic foveation for video compression using a neurobiological model of visual attention. IEEE Trans Image Process, 13: 1304–1318

Itti L, Koch C. 2000. A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Res, 40: 1489–1506

Itti L, Koch C, Niebur E. 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell, 20: 1254–1259

JTS. 2013. http://www.vividsolutions.com/jts/JTSHome.htm. accessed Nov 12, 2013

Kada M. 2002. Automatic generalization of 3D building models. Int Arch Photogramm Remote Sens Spatial Inf Sci, 34: 243–248

Kim D H, Yun I D, Lee S U. 2004. A new attributed relational graph matching algorithm using the nested structure of earth mover's distance. In: Proc of 17th Int Conf Pattern Recognition (ICPR'04)

Kim Y, Varshney A. 2006. Saliency-guided enhancement for volume visualization. IEEE Trans Vis Comput Graph 12: 925–932

Kolbe T H. 2008. Representing and Exchanging 3D City Models with CityGML in 3D Geo-Information Sciences. Berlin: Springer

Kopf J, Chen B, Szeliski R, et al. 2010. Street Slide: Browsing Street Level Imagery. ACM Trans Graph, 29: 96: 1–8

Lee C H, Kim Y, Varshney A. 2009. Saliency-guided lighting. IEICE Trans Inf Systems, E92.D: 369–373

Lee T. 2009. Robust 3D street-view reconstruction using sky motion estimation. In: Proceedings of IEEE Int Workshop on 3-D Digital Imaging and Modeling (3DIM2009) in Conjunction with ICCV

Mao B, Fan H, Harrie L, et al. 2010. City model generalization similarity measurement using nested earth mover's distance. In: Proceedings of 13th Workshop of the ICA commission on Generalisation and Multiple Representation, Zurich, Switzerland, 12-13 September

Mao B, Ban Y. 2011. Visualization of 3D city model through the internet using CityGML and X3DOM. Cartographica, 46: 109–114

Mao B, Harrie L, Ban Y. 2011. Detection and typification of linear structures for dynamic visualisation of 3D city models. Computs Environ Urban Syst, 36: 233–244

Mao B, Ban Y. 2013. Generalization of 3D building texture using image compression and multiple representation data structure. ISPRS-J Photogramm Remote Sens, 79: 68–79

Menzel N, Guthe M. 2010. Towards perceptual simplification of models with arbitrary materials. Comput Graphics Forum, 29: 2261–2270

Miao Y, Feng J. 2010. Perceptual-saliency extremum lines for 3D shape illustration. Visual Comput, 26: 433–443

Micusik B, Kosecka J. 2009. Piecewise planar city 3D modeling from street view panoramic sequences. IEEE Conf Comput Vision Pattern Recognition (CVPR), Miami, USA

Mortara M, Spagnuolo M. 2009. Semantics-driven best view of 3D shapes. Comput Graph, 33: 280–290

Nurminen A. 2008. Mobile 3D city maps. Comput Graphics, 28: 20–31

Parry M, Ribarsky W, Shaw C, et al. 2002. Organization and simplification of high-resolution 3D city facades. In: Proceedings of SPIE Aerosense Symposium, Visualization of Temporal and Spatial Data for Civilian and Defense Applications IV, 4744B

Rakkolainen I, Vainio T. 2001. A 3D city information for mobile users. Comput Graphics, 25: 619–625

Raubal M, Winter S. 2002. Enriching way finding instructions with local landmarks. Lecture Notes Comput Sci, 2478: 243–259

Salmen J, Houben S, Schlipsing M. 2012. Google street view images support the development of vision-based driver assistance systems. Intell Vehicles Symposium. 891–895

Schulte-Coerne T. 2005. Visualisierung von distanzfunktionen insbesondere der earth movers distance. GI Inf 2005, Schloss Birlinghoven: 8. bis 9. April 2005

Sester M, Brenner C. 2000. Typification based on Kohonen feature nets. The 1st International Conference on Geographic Information Science (GIScience 2000) Savannah, Georgia, USA

Sester M, Brenner C. 2004. Continuous generalization for fast and smooth visualization on small displays. Int Arch Photogramm Remote Sens, 34: 1293–1298

Frintrop S, Jensfelt P, Christensen H. 2006. Attentional landmark selection for visual SLAM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06), October 2006

Sun X, Qing W H, Hua Y W. 2009. Design and implementation of global 3D visualization client program on intelligent mobile devices. Third Int Conf Multimedia Ubiquitous Eng, 459–464

Thiemann F. 2002. Generalization of 3D building data. The Int Arch Photogramm Remote Sens Spatial Inf Sci, 34: 286–290

Thompson K G, Bichot N P. 2005. A visual saliency map in the primate frontal eye field. Prog Brain Res, 147: 251–262

Vincent L. 2007. Taking online maps down to street level. Compute, 40: 118–120

Wang M, Li J, Huang T, et al. 2010. Saliency detection based on 2D log-gabor wavelets and center bias. ACM Multimedia, 2010: 979–982

Winter S. 2003. Route adaptive selection of salient features. COSIT, 2003: 349–361

Xiao J, Fang T, Zhao P, et al. 2009. Image-based street-side city modeling. ACM Trans Graphics, 28: 114

Xj3D. 2013. http://www.xj3d.org/. accessed Nov 12, 2013

Yantis S. 2005. How visual salience wins the battle for awareness. Nat Neurosci 8: 975–977

Zhu Q, Li D, Zhang Y. 2002. Integration of DEMs, images and 3D models. Photogramm Eng Remote Sens, 68: 361–367