GENERALIZATION OF TILED MODELS WITH CURVED SURFACES USING TYPIFICATION

Richard Guercke [a], Junqiao Zhao [b], Claus Brenner [a] and Qing Zhu [b]

[a] Leibniz Universität Hannover, Institute of Cartography and Geoinformatics , Appelstr. 9a, 30167 Hannover, Germany – (richard.guercke, claus.brenner)@ikg.uni-hannover.de
[b] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, P.R. China – johnzjq@gmail.com, zhuqing@lmars.whu.edu.cn

KEY WORDS: Multi-Scale Representation of Spatial Data, Generalization, Building Models, Tiles, Roof, Curved

ABSTRACT:

Especially for landmark buildings or in the context of cultural heritage documentation, highly detailed digital models are being created in many places. In some of these models, surfaces are represented by tiles which are individually modeled as solid shapes. In many applications, the high complexity of these models has to be reduced for more x efficient visualization and analysis. In our paper, we introduce an approach to derive versions at different scales from such a model through the generalization method of typification that works for curved underlying surfaces. Using the example of tiles placed on a curved roof – which occur, for example, very frequently in ancient Chinese architecture, the original set of tiles is replaced by fewer but bigger tiles while keeping a similar appearance. In the first step, the distribution of the central points of the tiles is approximated by a spline surface. This is necessary because curved roof surfaces cannot be approximated by planes at large scales. After that, the new set of tiles with less rows and/or columns is distributed along a spline surface generated from a morphing of the original surface towards a plane. The degree of morphing is dependent on the desired target scale. If the surface can be represented as a plane at the given resolution, the tiles may be converted to a bump map or a simple texture for visualization. In the final part, a perception-based method using CSF (contrast sensitivity function) is introduced to determine an appropriate LoD (level of detail) version of the model for a given viewing scenario (point of view and camera properties) at runtime.

1. INTRODUCTION

1.1 Generalization

The research behind the results in this paper was done in the context of the generalization of building models. Especially roof and wall surfaces may be modeled by individual tiles where “tiles” refers not only to roof tiles but also to bricks in walls or any other basic unit of which a surface is constructed.

In the context of the generalization of building models, our approach can be used to produce different LoD models from an original model. The typification procedure described in this paper can be used in large-scale visualization if the representation of the surface by a plane with the tiles as textures and / or bump maps is too coarse, and the number of tiles and their geometric complexity make the rendering of all tiles an expensive and unnecessary operation.

The basic idea of the approach is to interpret the distribution of the tiles in the original model as a sampling of a surface. This surface is approximated by another surface – in our experiments, we used a Bézier spline surface. According to the desired resolution or the viewing scenario, this surface may be simplified – we implemented, for example, a linear transition from the original surface to a plane. On this new surface, enlarged and possibly geometrically simplified instances of the tile model are distributed along paths derived from an interpolation of the traces of the rows and columns of the original model on the original surface.

In order to optimally select these simplified models during real-time visualization, the contrast sensitivity function (CSF) is employed as a criterion. We evaluate the spatial frequency and the contrast of tiles to obtain the perceptibility of the tile pattern. Then the automatic LoD management is achieved.

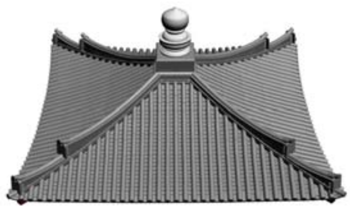

Figure 1. One tiled roof in Chi Lin Nunnery

In order to evaluate if our approach can provide suitable results for real-world data, we used a detailed model of a real roof surface from an ancient Chinese temple of the Chi Lin nunnery in Hong Kong (see Figure 1) for our experiments.

1.2 Related Work

Building generalization often models roof surfaces as planes (such as in [Kada, 2007]), and surfaces composed of individual tiles are not an issue because there are only very few models of such fine granularity. [Buchholz, 2006] describes an approach to procedurally derive textures for the visualization of 3D city models at different levels of detail. The issue of individually modeled tiles is, however, not addressed there either.

Human perception is introduced into polygon reduction in order to allocate more geometry to visually more important places [O'Sullivan et al., 2004]. The proposed methods can be divided into two kinds: one is implemented in model space and the other is in screen space. The former is usually by evaluating the perception of geometry using projected errors or curvature [Luebke et al., 2003; Winkler, 2000]. The latter evaluates the perception of an image on the screen by introducing more rigorous human vision system (HVS) models derived from research on image processing [Winkler, 2000]. However, most existing methods treat the perception information as weights to adjust the sequence of simplification operations such as edge collapse etc. [Qu and Meyer, 2008], which is not suitable for the aggregation of details. A new perceptually driven primitive location method was proposed to overcome this shortage by introducing the top-down constrain [Du et al., 2008]. However, following the idea of the previously proposed generalization framework [Guercke and Brenner, 2009], proper generalization methods for model parts with specific semantics are needed, such as the tiled roof model.

2. MODELING AND FEATURE EXTRACTION ISSUES

2.1 Introduction: A Pragmatic Approach

Generalization and feature extraction are closely related because generalization procedures can greatly benefit from or depend on semantic knowledge that may not be present in the data set to be processed.

As the general problem of feature extraction is in itself a wide field for research, we used heuristic approaches to extract the necessary semantic information for the generalization process. For many applications, these heuristics may work as well – perhaps with some small adaptations.

We tested our approach using the model of a roof surface from the Chi Lin Nunnery in Hong Kong [Li et al., 2006]. Especially in the feature extraction step, we were content with getting satisfying results for this data set. For other surfaces, more sophisticated methods may be necessary.

2.2 The Tile Model

In order to be suitable for the approach that is presented in this paper, the model should contain only a limited number of structurally different types of tiles with a clearly defined relationship.

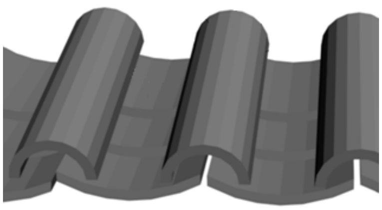

Figure 2. Placement of the upper and lower tiles

In the case of the roof, there are two different types of tiles: one for the upper (Tong Wa) and one for the lower layer (Ban Wa). Figure **2** illustrates the layout of the upper and lower tiles.

For each type of tile, a template tile is stored. This template tile stores the geometry of the tile in tile coordinates. The center of (the bounding box of) the tile is supposed to be the origin of the tile coordinate system. The x and y direction define the main directions of the tile, the z axis is the "normal" of the tile.

The distribution of the tiles is modeled by virtual tile objects that store a reference to the template tile, the position of the tile's center, the direction of its local coordinate axes in world coordinates, and scaling factors of the tile in its local x, y, and z directions. Using these parameters, the geometry of the tiles can be obtained by a simple linear transformation of the vertices in the template model.

2.3 Approximation of the Underlying Surface

Especially in the case of curved surfaces, the exact underlying shape is often the result of a combination of planning and aging. Therefore, there is usually no analytic model for the shape. For this reason, we approximate it by a Bezier spline surface because a representation based on control points is useful for the generalization process (for the simplification of the underlying shape described in section 3.2).

This surface was obtained by a least squares adjustment procedure minimizing the vertical distance between the centers of the tiles and the approximating spline surface. In order to reduce the complexity of the adjustment procedure, we used the fact that the surface could be described as a height function $z=f(x,y)$. This way, only the z values for the control points had to be estimated from the set of the central points of the tiles.

In our approach, we used C^2 continuous Bezier surfaces for the approximation to ensure smooth transitions between the patches. The borders between the patches that form the surface are aligned with the coordinate axes for simplicity.

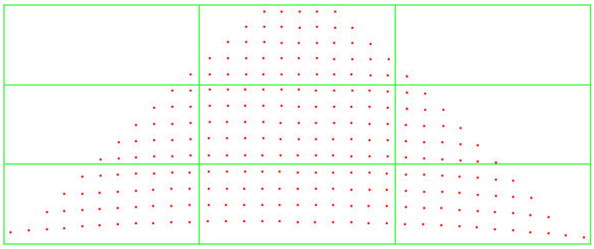

Figure 3. Boundaries for 3x3 Patches

The boundaries of the patches were chosen to be spaced equally in the range defined by the bounding box of the tiles' centers. In the adjustment process, the sum of the squared vertical distances of the tiles' centers to the surface was minimized.

Figure **3** shows the patch boundaries for three by three patches.

Because both versions yielded good results with maximum vertical distances of all tiles to the surface of less than one centimeter, we used only 2x1 patches for our experiments.

2.4 Identifying Rows and Columns

The generalization algorithm that is described in this paper relies on the tiles' being arranged in columns and rows. When the model is created, it is no problem to mark or organize the tiles in such a way that the rows and columns are marked explicitly.

For our sample data set, we had to detect the rows and columns from the distribution of the tiles. A general solution to this problem is a rather complex feature extraction task and beyond the scope of this paper.

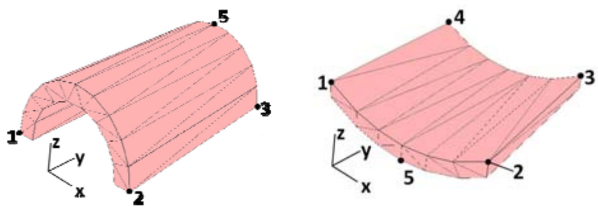

Figure 4: The upper (Tong Wa) and lower tiles (Ban Wa)

Figure 4 illustrates the distribution of characteristic vertices on the shapes of the different tiles. The characteristic vertices 1-5 in the given model could be determined by their distances to the center and their z-value. Using the characteristic points, the orientation of the tiles could be reconstructed. Since the rows and columns of the roof surface in our test data set were roughly aligned with the local x and y directions of the tiles, they could easily be identified once the orientation of the tiles had been established.

3. GENERATION OF LOD MODELS

3.1 Overview

The basic idea of the algorithm is to use a surface that approximates the original distribution of the tiles to place the tiles in the generalized model.

Depending on the desired resolution, fewer enlarged – and possibly geometrically simplified – instances of the tile models are distributed on an underlying surface which may be the original surface or – at lower resolutions – a simplified version of it. In our experiments, we used a surface resulting from a morphing of the original surface towards a plane.

In its current version, the algorithm uses the first (base) row to determine the number of columns, assuming that this is the widest row, containing the maximum number of tiles (columns).

We did not incorporate a simplification of the basic shape of the tiles into our approach. This would, however, offer considerable potential in a real application because the tile models may be quite complex – the lower tiles in our experiment had 28 vertices and the upper ones had 42 vertices which definitely means that there is potential for simplification at smaller scales.

The layers containing the different kinds of tiles were treated separately in our experiment – only the placing of the columns of the upper tiles was adapted to make the upper tiles cover the seams between the lower tiles properly.

3.2 Simplification of the Underlying Surface: Morph towards a Plane

The adjustment of the underlying surface serves several purposes. The first is a reduction of the geometric complexity of the underlying shape. Additionally, it offers a smooth transition to smaller scale models in which the tiles are modeled as bump maps or textures on a plane.

For our test case, we used a plane through the tiles on the ridge and the corner tiles in the base row. Although this plane is lying almost completely above the original shape, it has the advantage that the different roof surfaces are going to form a quite regular hipped roof when they are combined.

The control points of the spline are moved towards the plane in a linear way:

$$c_{i,new} = c_i + t \cdot (p(c_i) - c_i)$$

where c_i is the position of the i^{th} control point of the original spline, $p(c_i)$ the vertical projection of c_i on the target plane, and $c_{i,new}$ the position of the i^{th} control point in the spline representing the new underlying surface – for t=0, the surface is unchanged, for t=1, the surface is identical to the plane. According to the purpose of the generalization, appropriate values of t for have to chosen for the different LoD models.

In the following sections, the term "underlying surface" will refer to the morphed surface.

3.3 Building the Base Row

In a first step, a Bezier curve y=b(x) approximating the distribution of the centers of the original tiles in the first row is calculated.

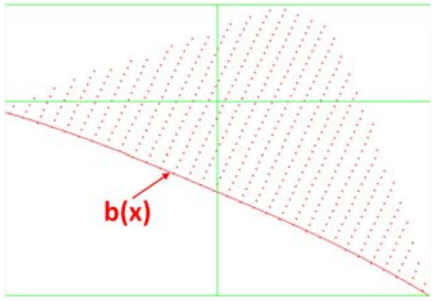

Figure 5. Trace b(x) of the base row in the (x,y)-plane

Figure 5 shows the distribution of the original tiles in the (x,y)-plane and the approximation spline curve b(x) through the centers of the tiles in the first row.

The restriction that b should be a function of x is introduced by the approximation process rather than the generalization algorithm. In fact, any function defining a surface curve B(t) on the underlying surface s=s(x,y) can be used. This function B may be defined directly or through function c: (x,y)=c(t) with B(t) = s(c(t)). In our example, the control parameter t is the x-value, so we can set

$$B(t) = B(x) = s \cdot \binom{x}{b(x)}.$$

The distribution of the tiles is done in parameter space (on the x axis). In the current version, the x values for the tiles t_i were evenly between the positions of the first and last tiles in the original first row: All tiles have the same distance of Δx in x direction with

$$\Delta x = \Delta x_{ori} \cdot \frac{k}{k_{ori}}$$

where k_{ori} is the original number of columns, k is the desired number of columns, and Δx_{ori} the average distance between the original tiles on the x axis. For the roof model in our testing scenario, this yielded acceptable results but in general, it is a problematic approach because it does not take the curvature of the surface into account.

A better solution is to use the actual distances on the surface for the distribution of the tiles. In the first step, an initial scaling factor $f=k/\ k_{ori}$ is defined. The first tile is placed in a starting position. The parameter t_0 of this first tile has to be known. The initial position of the next tile is determined as

$$\vec{x}_{i,step=0} = B(t_{i-1} + f \cdot \Delta t_{ori})$$

where t_{i-1} is the parameter of the previous tile and Δt_{ori} the average difference of the parameters for neighboring tiles in the base row of the original model. $t_{i,step=0} = f \cdot \Delta t_{ori}$ is the initial value of the curve parameter t for the current tile i.

After that, the error is calculated:

$$e = |\vec{x}_{i,step} - \vec{x}_{i-1}| - d_{tile}$$

with d_{tile} being the average distance between tile centers in the base row scaled by f, $\vec{x}_{i-1}$ the position of the preceding tile, and $\vec{x}_{i,step}$ the current position of the current tile. If |e| is smaller than a threshold value (depending on the scale), the tile position can be accepted. If e > 0, then the actual distance between the two tiles is too large. In this case, $t_{i,step}$ has to be reduced by some amount. If e < 0, then the distance between the tiles is too small and $t_{i,step}$ has to be increased.

The process terminates when either *k* tiles have been distributed or if a tile would have to be placed outside of a given boundary. A simple boundary condition might be that the center of the last tile should not be beyond the center of the last tile in the original base row – with some buffer to allow for the increased size of the tiles.

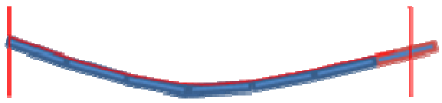

Figure 6. Tiles in the base row are too large

It may happen that due to the curvature of the underlying surface, it is not possible to place k tiles in the base row or that after placing k tiles, there is a gap to the intended end of the row. In the first case, the tiles are too big and the process for building the base can be repeated with a decreased scaling factor f for the tiles. If there is a gap, the tiles are too small, and the process is repeated with an increased scaling factor f. In Figure 6, the last tile could not be placed because it would fall outside the range specified for the tiles. The rectangles represent the tiles, and the bars on the left and right side are the boundaries beyond which no tile should extend.

3.4 Distribution of the Tiles on the Surface

The distribution of the tiles along the columns is done in a similar way to the building of the base row: the first tile of column i is the i^{th} tile in the base row, and the tiles are distributed along the surface curve defined by the trace of the column. The scaling factor f is taken from the previous step.

In our sample data set, all columns are parallel in the (x,y) plane. In the more general case, one can approximate the traces of the columns by spline functions and perform an interpolation of the parameters to get the traces of the new columns. The end of the columns can be established by interpolating the ends of the original columns.
The main directions of the tiles were aligned with the direction of the columns: The local y axis of the tiles pointed in the direction of the traces of the columns. This property was preserved in the distribution of the new tiles in the LoD models:

$$\vec{y} = \vec{c}_{i+1} - \vec{c}_i + \vec{v}_{off}$$

where y is the direction of the tile's local y axis in world coordinates, c_{i+1}-c_i the vector from the center of the current tile to the center of the next one, and v_{off} an offset vector that is introduced to model the tilting of the tiles to avoid overlaps. The x vector can be obtained as the cross product of y and the global up vector $(0,0,1)^T$. The normal z of the tile is defined as the cross product of x and y. All local direction vectors of the tile are normalized to make them form an orthonormal basis.

In our experiments, we applied the scaling factor f only along the local x and y directions of the tiles to support a smooth transition towards a plane. If the height of the tiles is scaled as well, then v_{off} will also have to be scaled.

4. PERCEPTION-BASED SELECTION OF AN APPROPRIATE LOD MODEL

Using the established generalization method above, a number of LoD models for a tiled roof can be generated automatically. However, how to optimally select the right model for real time visualization is still an unsolved question. It's understandable to leave it to the user to set the switch distances for each LoD. But this trial and error process is obviously too time-consuming. Moreover, the improper selection of LoD would cause heavy popping effects which decrease the efficiency of transmitting both apparent and semantic information of such kind of models.

A number of perceptually driven methods had been proposed in recent decades [Luebke et al., 2003]. Reddy firstly introduced the principle perceptual model into the LoD selection issue [Reddy, 1997]. The spatial frequency of objects was analyzed by image segmentation using rendered images from multiple viewpoints. If the spatial frequency difference of two LoD models is above (or below) the visual acuity, a coarser (or finer) LoD is to be selected. Similar works have been done in [Luebke and Hallen, 2001] and [Cheng et al., 2006]. However, most of the existing methods evaluate the HVS factors using the curvature of vertices or faces, which obviously does not fit the component structured-model i.e. tiled roof or walls made of bricks. Therefore, a new perceptually based LoD selection method is needed for our test model.

As the most important component of HVS, CSF describes the quantified relationship between the visual perception and the factors of spatial frequency and contrast threshold, as illustrated in
Figure **7**. The expression of CSF is as follows:

$$A(\alpha) = 2.6(0.0192 + 0.144\alpha)\exp(-(0.144\alpha)^{1.1})$$

where $A(\alpha)$ is the contrast threshold of spatial frequency α (c/deg). If the current contrast is lower than $A(\alpha)$, the signal is invisible.

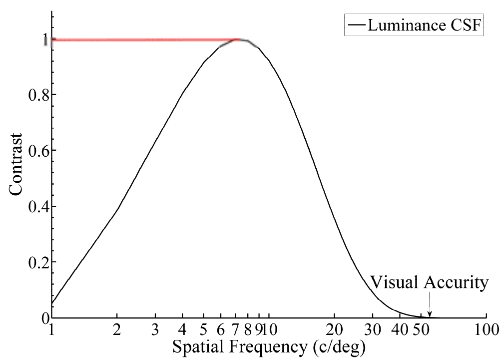

Figure 7. The CSF

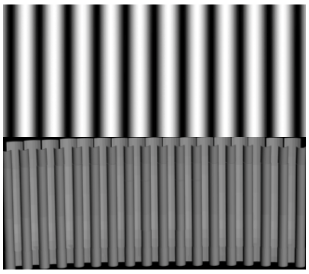

Figure 8. The grating signals (Upper: the theoretically sine wave. Lower: the top view of tiled roof)

Because the upper tiles hide the seams between lower tiles, we can treat the lower tile as the uniform background while the upper tile as the foreground, the represented pattern, as shown in Figure 8, matches the contrast grating signal which is used to evaluate the CSF quite well. Based on this fact, an approximated CSF model is proposed by evaluating the contrast and spatial frequency of roof pattern: For contrast, the luminance of tile depends on the normals of the tile, so it can be represented as:

$$L = k\cos\left(\frac{\pi}{2} - \theta\right) = k\sin\theta$$

Where L is the luminance of a tile, θ is the angle between the normal vector and the direction of light, k is the coefficient of simple illuminate model obeying Lambert's cosine law. Because the shape of the upper tile is a kind of semicircle which would likely to have both the highest luminance as well as the lowest luminace at most of the viewing angles, as illustrated in Figure 9. The Michaelson contrast of tiled roof, defined as (Lmax−Lmin)/(Lmax+ Lmin), can be calculated by:

$$C = \frac{\sin(min(\theta)) - \sin(max(\theta))}{\sin(min(\theta)) + \sin(max(\theta))}$$

where θ is the angle between the direction of light and the normal vectors of the triangles of the upper tile.

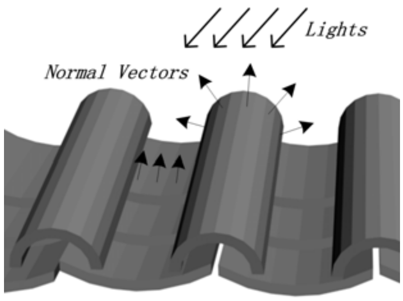

Figure 9. The computing of contrast

For spatial frequency, the projected distance d_{prj} between vertices 5 can be treated as the wave length of the contrast grating signals represented by the upper tiles. Therefore,

$$f = \frac{1}{d_{prj}}$$

where f is the spatial frequency evaluated at d_{prj}, the projected distance between upper tiles on screen. The CSF can then be rewritten as follows:

$$A(d_{prj}) = 2.6\left(0.0192 + 0.144\frac{1}{d_{prj}}\right)exp\left(-\left(0.144\frac{1}{d_{prj}}\right)^{1.1}\right)$$

In order to find the best LoD model for a given viewing scenario, the d_{prj} between two adjacent columns is an important criterion. At a given viewing position during real time visualization, we firstly find the nearest upper tiles in two nearby columns. d_{prj} can then be calculated by:

$$d_{prj} = \frac{Z \cdot {d_1}^2}{\tan(\alpha)} + \frac{Z \cdot {d_2}^2}{\tan(\beta)}$$

where z is the distance from the camera to the nearest clipping plane, d_1 and d_2 are the distances from vertices 5 to the viewing orientation, α, β are the angles between vertices 5 to the viewing orientation, as shown in Figure **10**.

If the realtime evaluated contrast threshold $A(d_{prj})$ is lower than C, a LoD model containing a larger d is to be selected. It is to be noted that if other factors like velocity as well as eccentricity are to be taken in consideration, a more sophisticated HVS model is needed.

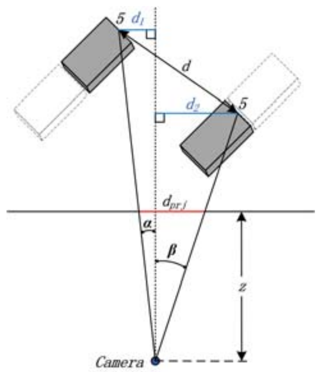

Figure 10. Computing of projected d

5. EXAMPLES

Figure 11 shows the original model and detailed views of parts of the roof in different LoDs. The first detail drawing shows a part of the roof from the model with 32 rows – the same number of rows as in the original model. One can see that the tiles fit neatly without noticeable gaps or overlaps.

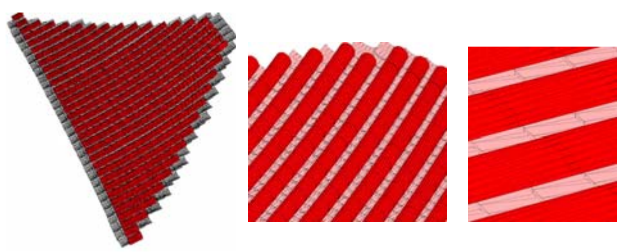

Figure 11. The original roof surface and details from different LoD models

There is some overlap between the upper tiles because the tiles were distributed along the columns without compensating for the loss of length in the columns resulting from the transition towards the plane. The effect could have been avoided had the iterative approach described in section 3.3 been used. In this case, the effect is only visible in the wireframe model from close range, so it can be accepted for visualization purposes.

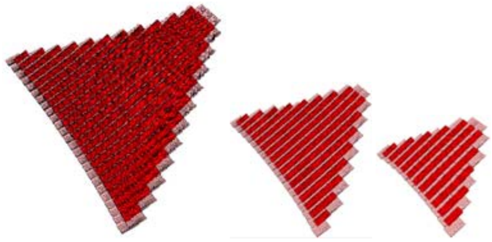

Figure 12. LoD models with 32, 25 and 12 tiles in the base row

Figure 12 shows LoD models for 32, 25, and 12 tiles in the base row. The corresponding values for d (see section 4) are 27, 34, and 76 centimeters.

6. CONCLUSION AND OUTLOOK

We have presented ideas for the generalization of tiled surface models and their application to the model of a roof surface of an ancient Chinese temple.

Though more research is definitely necessary to get a precise understanding of the potential and limitations of our approach, the results from the real-world data set are promising.

The main purpose of the experiments described in this paper was to test if our ideas are suitable for real-world situations. Especially in the context of generalization for visualization, we think that we could produce promising results.

Concerning the iterative algorithm for the distribution of tiles along a surface curve, a closer inspection of the potential and limitations is necessary.

At smaller scales, using simplified tile models can reduce the complexity of the model considerably without causing a noticeable loss of quality. Because the tiles are enlarged in the process, the selection of an appropriate simplified tile model has to be done with care. In our experiments, we did not introduce simplified tile models.

In this paper, we described how Contrast Sensitivity Function (CSF) could be used to select an appropriate LoD from a set of such models that were derived in advance. Recently, a novel perceptually based method for planning the discrete LoD for complex model façade was introduced [Zhu et al., 2009]. By using the idea, the visual model could also be employed in the process of building the LoD models to select values for the morphing parameter t depending on the desired number of columns k and the original surface.

For applications beyond visualization, techniques for the detection and avoidance of gaps or overlaps between the tiles may be necessary. Especially when different layers or neighboring surfaces have to be aligned, these problems can cause irritating effects.

ACKNOWLEDGEMENTS

The research for this paper was funded by the German Federal Ministry of Education and Research (BMBF) in the context of the GDI-Grid project and by the National Basic Reseach Program of China (No. 2010CB731800) as well as the National High Technology Research and Development Program of China (No. 2008AA121600).

REFERENCES

Buchholz, H., 2006. Real-time Visualization of 3D City Models. PhD thesis, Universität Potsdam.

Cheng, I., Shen, R., Yang, Y. and Boulanger, P., 2006. Perceptual Analysis of Level-of-Detail: The JND Approach, Proceedings of the Eighth IEEE International Symposium on Multimedia: 533–40.

Du, Z.Q., Zhu, Q. and Zhao, J.Q., 2008. Percetion-driven simplification methodology of 3D complex building models, ISPRS2008, Beijing.

Guercke, R. and Brenner, C., 2009. A Framework for the Generalization of 3D City Models, Proceedings of 12th AGILE Conference on GIScience.

Kada, M., 2007. 3D Building Generalisation by Roof Simplification and Typification, Proceedings of the 23rd ICC.

Lee, C.H., Varshney, A. and Jacobs, D.W., 2005. Mesh saliency. ACM Transactions on Graphics, 24(3): 659-666.

Li, D.R., Zhu, Y.X., Du, Z.Q. and Hong, T., 2006. Virtual Tang-Style Timber-frame building complex. Lecture Notes In Computer Science, 4282:880-888.

Luebke, D. and Hallen, B., 2001. Perceptually Driven Simplification for Interactive Rendering, Proceedings of the Eurographics Workshop. Rendering Techniques 2001. Springer, London, United Kingdom, pp. 223-234.

Luebke, D. et al., 2003. Level of Detail for 3D Graphics. Morgan Kaufmann, San Francisco.

O'Sullivan, C., Howlett, S., McDonnell, R., Morvan, Y. and O'Conor, K., 2004. Perceptually adaptive graphics. Eurographics State of the Art Reports: 141-164.

Qu, L. and Meyer, G.W., 2008. Perceptually guided polygon reduction. IEEE Transactions on Visualization and Computer Graphics, 14(5): 1015-1029.

Reddy, M., 1997. Perceptually Modulated Level of Detail for Virtual Environments, University of Edinburgh, Edinburgh, United Kingdom.

Winkler, S., 2000. Vision models and quality metrics for image processing applications, Swiss Federal Inst. Technol, Lausanne, Switzerland

Zhu, Q., Zhao, J.Q., Du, Z.Q. and Zhang, Y.T., 2009. Quantitative analysis of discrete 3D geometrical detail levels based on perceptual metric. Computers and Graphics, 34(1):55-65.