MULTI-RESOLUTION REPRESENTATION OF DIGITAL TERRAIN AND BUILDING MODELS

Fuan Tsai, Wan-Rong Lin and Liang-Chien Chen

Center for Space and Remote Sensing Research
National Central University
Zhong-li, Taoyuan 320 Taiwan
ftsai@csrsr.ncu.edu.tw

KEY WORDS: Multi-Resolution, Digital Terrain Model, Building Model, Level of Detail, Cyber City

ABSTRACT:

This research develops effective algorithms for multi-resolution representations of three-dimensional (3D) digital terrain and building models to achieve better performance in cyber city applications. The objective is to create multiple levels of detail (LOD) of terrain meshes and polyhedral building models so they can be used efficiently according to viewing parameters and application requirements, while preserving critical features of the datasets. For terrain meshes, a tile-based approach is adopted. A mesh refinement algorithm based on modified quad-tree process is developed to generate multi-resolution representations of each terrain patch. On the other hand, a divide-and-conquer strategy is employed for the generalization of 3D building models to formulate LOD representations of complicated buildings. The idea is to apply generalization in 2D orthographic views of original polyhedral building models and then reconstitute simplified 3D models accordingly. Experimental results with complicated terrain and building datasets demonstrate that the developed LOD algorithms can improve cyber city performance significantly.

1 INTRODUCTION

Terrain and building are fundamental and two of the most important components in cyber city and other three-dimensional (3D) Geographic Information Systems (GIS) implementations. However, the vast amount of data in a large-scale cyber city modeling often poses a great challenge to efficient processing, analysis and visualization, especially in real-time applications. Level of Detail (LOD) is a commonly adopted technique to generate multi-resolution representations of objects in computer graphics and visualization (Luebke et al., 2003). The OpenGIS® CityGML (City Geography Markup Language) encoding standard proposed by OGC (Open Geospatial Consortium) also defines five levels of detail (LOD0~LOD4) for 3D digital city implementations (Gröger et al., 2008). However, the CityGML LOD specification is designed primarily based on functionality and thus may not be adequate in terms of performance consideration, especially for real-time visualization and applications. To address this issue, this paper presents systematic approaches to generate multi-resolution representations of large-scale digital terrain and building models that can be used to improve the performance in data transmission, processing and rendering of a cyber city system.

2 LOD FOR TERRAIN MESHES

As there are usually millions of points and polygons in a large-scale digital terrain model, it is a practical necessity to reduce the data amount for efficient processing and rendering. Level of detail techniques have been proposed for multi-resolution representation of terrain meshes, such as Bin-tree hierarchies (Blow, 2000), Bin-tree regions (Cignoni et al., 2003), geometric clipmaps (Losasso and Hoppe, 2004) and Quad-tree based meshes refinement (Tsai et al., 2006). Among them, Quad-tree based approaches are more suitable for geo-spatial applications, because they can better preserve critical terrain features while reduce the data amount.

A previous study suggested applying Quad-tree simplification on terrain meshes separated into tiles and proposed an adaptive progressive mesh data structure to achieve near real-time rendering frame rate for complicated digital terrain models (Tsai et al., 2006). However, in their algorithms the thresholds for quad-tree subdivision were determined from the difference between the maximum and minimum elevations of a tile and may cause over-sampling or under-sampling in areas. A new thresholding scheme (Tsai and Chiu, 2008) based on view-dependent image-space error metric was proposed to provide better LOD generation of terrain meshes. By calculating the ground sampling distance (GSD) as illustrated in Fig. 1 and Eq. (1) and (2) under different viewing parameters, different thresholds of quad-tree processing can be determined more reasonably according to "view-importance", thus generating more appropriate LOD datasets.

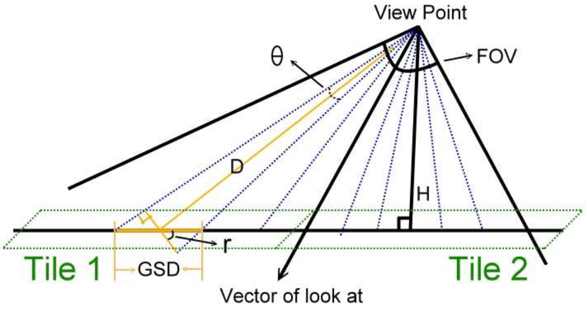


Figure 1: Ground sampling distance (GSD)

$$\theta = \frac{FOV}{\text{pixels per scanline}} \tag{1}$$

$$GSD = \frac{D\tan(\theta)}{\cos(r)} \tag{2}$$

In addition to the thresholding scheme, a nested LOD pre-process was also proposed (Tsai and Chiu, 2008). The idea was to generate an Outer-LOD-Set based on the coarsest level of the primary LOD terrain meshes (Core-LOD-Set). This can further increase the system performance in visualization, especially during the initializing stage of a large-scale application.

The reason of applying tile-based approach is that it is easy to implement view-dependent visualization for reducing the amount of data to process and render. However, a disadvantage in tile-based terrain visualization is the cracks caused by discontinuities (T-junctions) among tile boundaries. A commonly adopted technique to compensate this artifact is applying a pseudo generic texture layer beneath the terrain surface (Pouderoux and Marvie, 2005). Nevertheless, other than not providing true textures, this workaround will not work if the view angle is too low. A mesh refinement procedure was developed to address this issue. Taking Fig. 2 as an example, the procedure to remove T-junctions is listed in Algorithm 1.

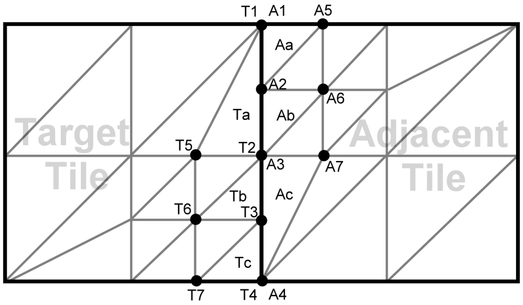


Figure 2: T-junction removal

Algorithm 1 T-junction removal procedure

1. Starting from $T1$ and $A1$, because $T1 = A1$, they remain unchanged.
2. Moving to the next pair, because $T2 \neq A2$, there exists a T-junction.
3. Remove Ta and add two new triangles, $\Delta(T1, T5, A2)$ and $\Delta(A2, T5, T2)$.
4. Continue the process from $T2$ and $A3$ and a new T-junction is found at $T3$.
5. Replace Ac with $\Delta(A3, T3, A7)$ and $\Delta(A7, T3, A4)$.

Combining these algorithms, the proposed tile-based quad-tree processing of terrain meshes can generate multiple LODs of large digital terrain datasets based on different viewing parameters. The algorithms can reduce data to process and render while preserving important terrain features. In addition, the adaptive data structure enables progressive transmission of data streams. All together, they provide high-performance terrain visualization capability for real-time applications.

3 MULTI-RESOLUTION 3D BUILDING MODELS

Unlike terrain data, most 3D building models used in cyber city implementations are not in mesh format. Therefore, mesh-based LOD schemes may not be applied to the process of building models. A few approaches have been proposed to deal with the LOD generation of 3D building models. For example, an algorithm based on mathematical morphology and curvature space of scale-spaces theory was presented to generalize 3D building models (Mayer, 1998). The algorithm was further refined by moving parallel facets toward each other to eliminate protrusion and close the gaps (Forberg, 2007). Another type of approach is segmenting building models into several structural elements and performing generalization on individual building segments, such as applying half-space modeling by cell decomposition and primitive instancing (Kada, 2007) or simplifying 2D projections of 3D geometries but only with linear and neighboring building groups (Anders, 2005).

Most of existing LOD techniques for 3D building models are either computationally expensive or are limited to certain types of buildings. A semi-automatic generalization approach is proposed to provide better multi-resolution representation of complicated building models. The idea is to apply generalization in 2D orthographic views of individual buildings and then reconstruct simplified 3D models accordingly as illustrated in Fig. 3. The process can be repeated with different generalization parameters so multiple levels of details can be created. The principle is similar to Anders (2005), but the procedure and algorithms employed in this study is very different. Anders (2005) utilized a program (CHANGE), which was originally designed to aggregate two-dimensional building ground plans for the generation of topographic maps, to simplify three projections of building groups and then grue them to form 3D block models. This approach is efficient for linear building groups but might not be adequate for complex buildings. On the other hand, the algorithms developed in this study is a divide-and-conquer strategy that is capable of dealing with building models with complicated structures and shapes. In addition, the developed algorithms can also generalize buildings with non-orthogonal façades, which are difficult to handle using existing methods.

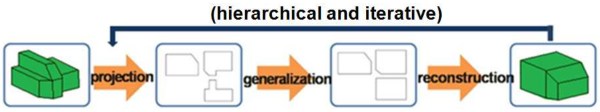
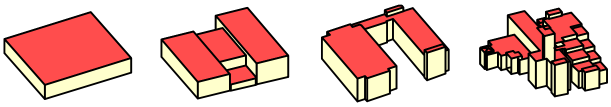


Figure 3: LOD for 3D building models

Before applying the generalization process, a geometric structure analysis is conducted to determine the complexity of each building and its number of levels to generalize. The shape complexity is defined as Eq. (3), where N_r is the number of vertices of the roof polygon and N_C is the number of vertices of the 3D convex hull of all roof structures. A few examples of shape complexity are demonstrated in Fig. 4. Based on calculated shape complexity, necessary levels of detail for individual building models can be determined. For instance, the most complicated model (SC=0.72) in Fig. 4 may require four levels of generalization while the second case (SC=0.3) may need only two levels.

$$SC = \frac{N_r - N_C}{N_r} \tag{3}$$



sc= 0 sc= 0.3 sc= 0.51 sc= 0.72

Figure 4: Shape complexity examples

After determining the levels to generalize, at least three orthographic projections of each building are generated and converted into raster formats. Then, a series of morphological operations (including dilation, point in polygon elimination, and erosion) is applied to create raster versions of the orthographic projections for constructing the outlines of building models as demonstrated in Fig. 5. A topology connector, which is modified from the Target Defined Ground Operator (TDGO) (Chen and Lee, 1992), is developed in this study to establish topological relationships among projection points. The modified TDGO performs topological encoding of each pixel according to the surrounding pixels in a 3x3 moving window. Some examples of TDGO encoding are listed in Fig. 6. Applying these operators to examine the generated raster projections, their edges and corner points can be identified correctly as displayed in Fig. 7.

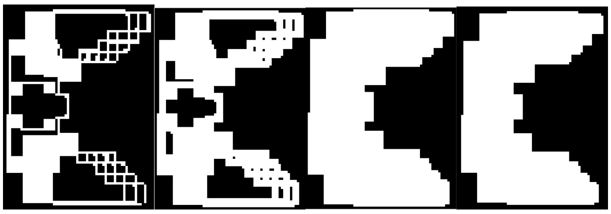


Figure 5: Outline generation of raster projections (left to right: (a) original projection; (b) dilation; (c) point-in-polygon filtering; (d) erosion)

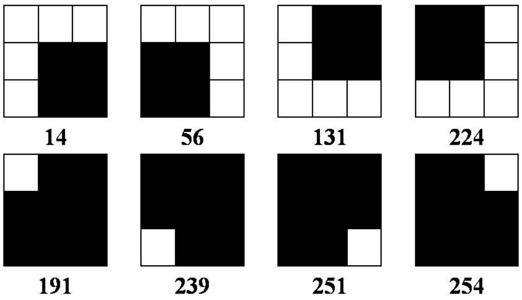


Figure 6: TDGO encoding examples

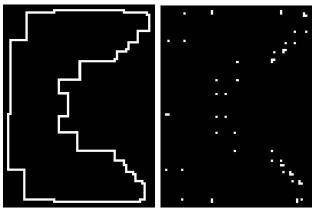
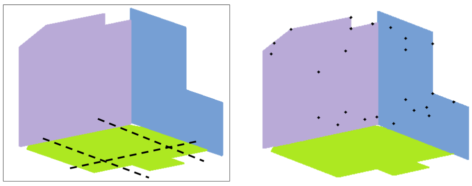


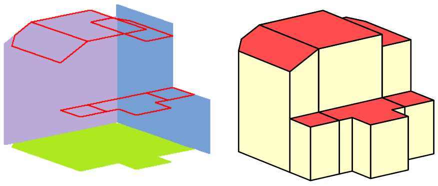
Figure 7: Identified edges and corners

Each orthographic projection is then generalized with convex-concave structure generalization and edge regularization. Convex-concave structure generalization is to detect convex and concave structures and eliminate small structures. Edge regularization is to eliminate short-length edges. In this step, the lengths of edges and the angle between the neighboring edges of a vertex in orthographic views are calculated based on the topological relationship of orthographic views. Concave and convex structures and edges with short lengths are detected by calculated angles and lengths. The areas of convex and concave structures are calculated and compared with a pre-defined threshold to remove minor structures. Edges with lengths smaller than pre-defined thresholds are also eliminated. New junction points for each eliminated structure or edge are calculated accordingly to reshape the orthographic projections.

After generalization of orthographic projections, simplified 3D models can be reconstructed from them as illustrated in Fig. 8. First, according to vertical orthographic views (front and side views), the horizontal orthographic view (top view) is segmented into several pieces. A loop tracing technique is applied to connect segmented pieces as polygons. After collecting all the segmented polygons, the heights of vertices points on segmented plans are calculated according to vertical orthographic views. These elevated points are reconnected in 3D space to shape the simplified 3D model.



(a) plan segmentation (b) height calculation



(c) 3D shaping (d) result

Figure 8: 3D model reconstruction from orthographic projections

For non-planar roof structures, they are detected and segmented in the geometric structure analysis step. As displayed in Fig. 9a, gabled roof structures are simplified according to the generalized plan view. When the ground plan is generalized, the topmost polygons are updated by transformation relationship through back-projection from ground plan to topmost polygons. For barrel roofs, the curve-shaped roof structure is maintained by curve fitting technique, as shown in Fig. 9b and placed on top of the major façades. There are different methods to fit a curve from points (Gallier, 2000). If there are sparse roof vertices, which is usually the case in polyhedral building models, a simple way is to use the two boundary vertices and the highest roof point for a conic arc fitting as illustrated in Fig. 9b. However, the boundary vertices need to be adjusted according to the generalization result of façades. This way, important characteristics such as building height and dimensions can be preserved. For complicated curves, collecting all vertices and performing a spline fitting or by least squares fitting (Coope, 1993) is a more appropriate approach, but requires more computation.

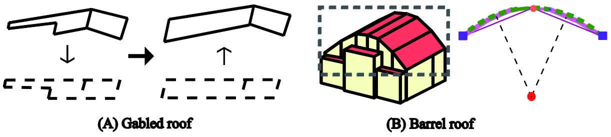
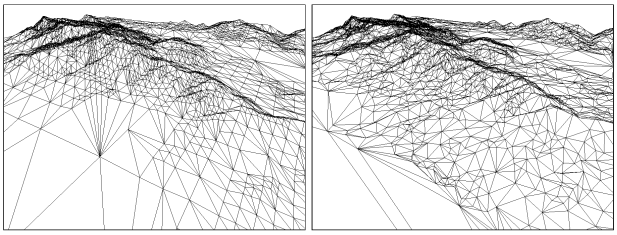


Figure 9: Generalization of roof structures

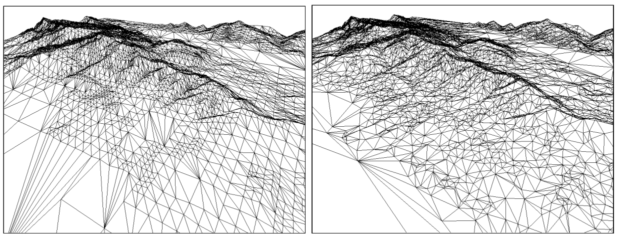
The described generalization method is highly automated and can deal with a variety of building models. However, in some special cases, additional (interactive) processes may be necessary. For example, for buildings with courtyards, the orthographic views of their inner structures will be detected interactively and then generalized with the generalization process of orthographic views. Another example is buildings with non-planar façades. To keep their curved characteristics, the non-planar façades are segmented from the original building models and generalized similar to the barrel roofs.

4 EXPERIMENTAL RESULTS

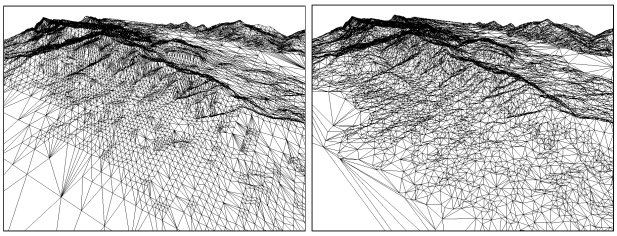
The developed multi-resolution representation algorithms for digital terrain and building models were applied to real datasets to validate their performances. Figure 10 shows the LOD1, 3 and 5 of a large terrain mesh and the comparison with Delaunay triangulation. Both the proposed quad-tree based LOD processing and Delaunay triangulation can effective reduce the data in different LOD levels.



(a) LOD1 (3087 vertices) (b) Delaunay LOD1



(c) LOD3 (5808 vertices) (d) Delaunay LOD3



(e) LOD5 (14900 vertices) (f) Delaunay LOD5

(g) Textured LOD1 (h) Textured Delaunay LOD1

Figure 10: Terrain LOD with proposed method (left) and Delaunay triangulation

Although it may appear that Delaunay has better triangulated meshes, the proposed method can also preserve important terrain features in different LODs. (If the meshes are textured, the difference between the two is almost indistinguishable, even in the low resolution LOD1 as displayed in Fig. 10g and h.) More importantly, the data structure of the proposed Quad-tree processing is more organized than Delaunay and makes the rendering more efficient. Taking T-junctions removal as an example, Delaunay will require significantly more efforts to remove T-junctions because the triangles on tile edges are irregular. In addition, for Delaunay triangulations, it will be inefficient to use the "difference vectors" scheme because vertices in different levels of Delaunay-based LOD do not have an "add-on" property. Therefore, it will be difficult, if not impossible, to achieve progressive transmission and adaptive rendering of Delaunay-based LOD tiles and thus inadequate for real-time visualization applications.

Figure 11 shows the effect of T-junction removal. From the figure, the crack at the tile boundary has been repaired with the proposed mesh refinement algorithm and resulting in a seamless terrain scene. The proposed mesh refinement algorithm is very efficient. In previous tests (Tsai and Chiu, 2008), the CPU time used to remove T-junctions at run-time was almost negligible and caused insignificant impact to the rendering performance. Figure 12 shows the comparison of accumulated CPU time for a fly-through of a large terrain (originally 5022 by 9555 grids and partitioned into 10 by 19 tiles) with and without performing T-junction removal while rendering the terrain meshes. It is clear that applying T-junction removal has increased very little rendering time, but the improvement in visual quality is significant as demonstrated in Fig. 11.

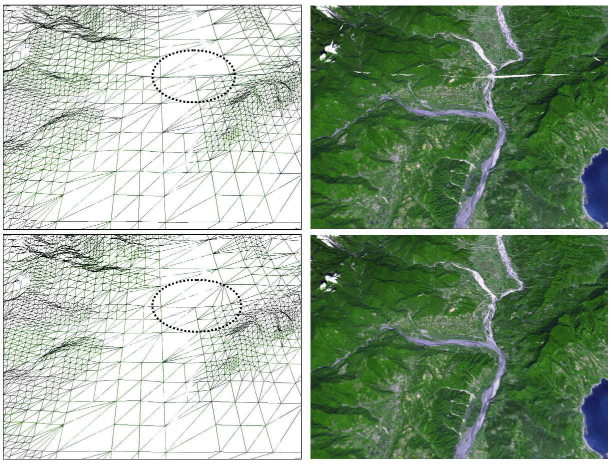


Figure 11: T-junction removal

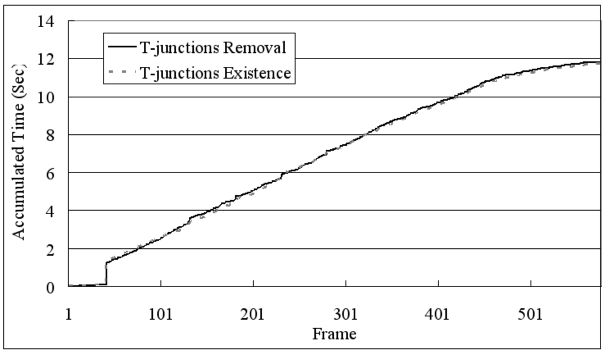
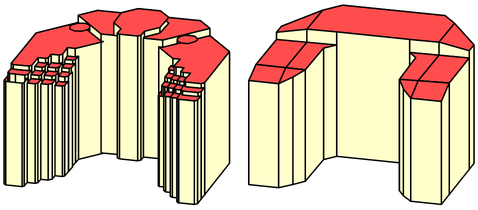


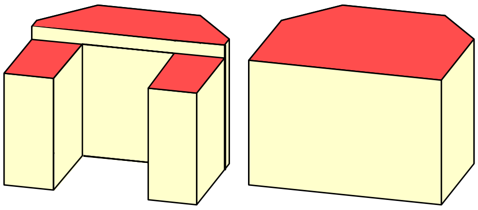
Figure 12: Cumulative rendering time comparison of T-junction removal

The proposed terrain LOD generating algorithms were tested with two large DEM datasets in different scenarios (Tsai and Chiu, 2008). From the tests, the frame rate of rendering can achieve at least 24.5 FPS (frames per second) within two-pixels error on screen display even in a very complicated mountainous terrain. This should be adequate for real-time visualization and applications.

Figure 13 displays an example of generated multiple building level of detail (BLOD) from a fairly complicated 3D building model. In this example four different levels of detail were generated with the generalization algorithms described above. From the figure, it is clear the generalization has effectively reduced the number of polygons from 290 to 61, 19 and 7 subsequently. Although the data (polygon) amount has been reduced significantly in lower-resolution building models, the characteristics of the building has been preserved.



(a) BLOD3 (290 polygons) (b) BLOD2 (61 polygons)



(c) BLOD1 (19 polygons) (d) BLOD0 (7 polygons)

Figure 13: Multiple LOD of building model

The example in Fig. 13 demonstrates that the proposed generalization algorithms and the iterative building level of detail generation procedure are effective for buildings with regular (planar) façades regardless their complexity. Figure 14 and 15 demonstrate generalization results of a few buildings with irregular shapes and structures including non-planar roof structures. These examples indicate that the developed BLOD algorithms are also effective for special building models.

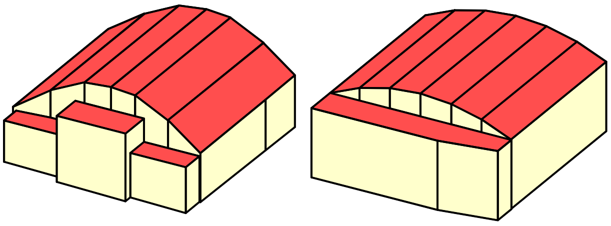


Figure 14: BLOD for special building model with barrel roof

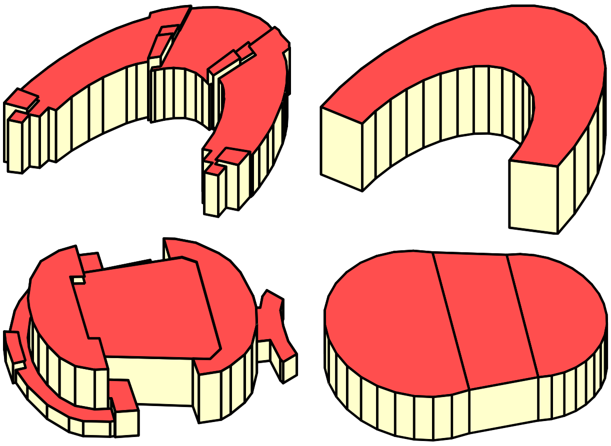
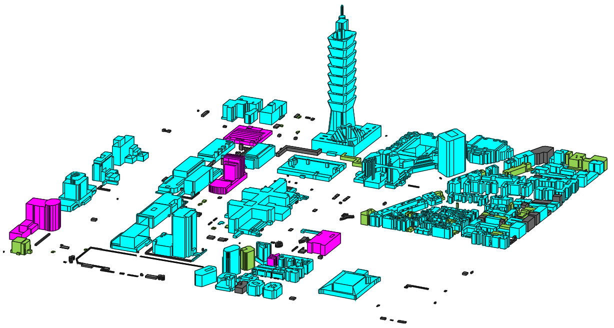
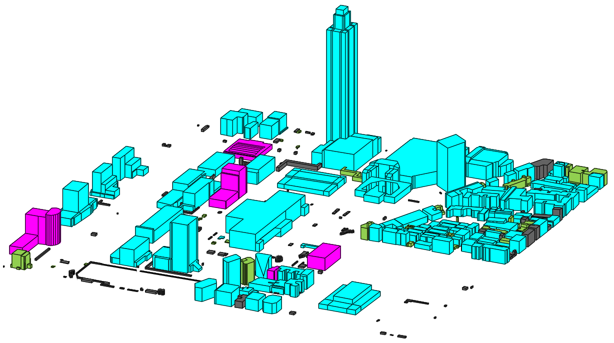


Figure 15: BLOD for building models with special shapes and structures

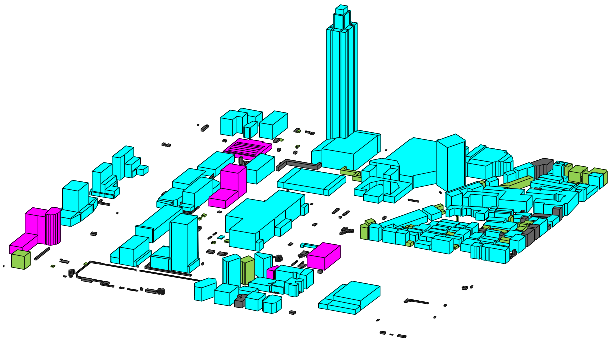
Applying the developed generalization algorithms to buildings in a city model, multi-resolution representation of the building models can be generated effectively. Figure 16 shows four levels of detail of a business district in Taipei. There are a couple of hundreds buildings with different degrees of complexity and styles in this area. The proposed algorithms generalize them according to determined shape complexity effectively. The generalization reduce the data amount significantly (from 27% to 38% of the original number of polygons as listed in Table 1) but still preserve important geometric characteristics (features) of buildings.



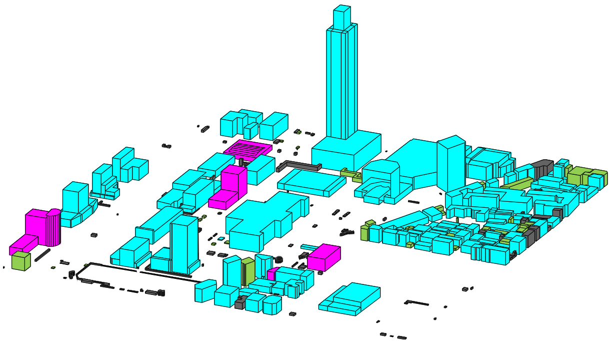
(a) BLOD3



(b) BLOD2



(c) BLOD1



(d) BLOD0

Figure 16: BLOD of a business district in Taipei Taiwan

With the generated multiple levels of detail, a cyber city system can load require building models progressively from BLOD0 to BLOD3 according to different viewing parameters and system or analysis requirements to increase the performance in visualization and analysis.

Table 1: Data reduction rate of BLOD

	BLOD3	BLOD2	BLOD1	BLOD0
Points	61795	22825	17535	16110
Reduction rate		37%	28%	26%
Polygons	14045	5350	4175	3841
Reduction rate		38%	30%	27%

One thing to note is that it seems data reduction is more aggressive when generalizing the models from BLOD3 to BLOD2 than the rest generalization. This might caused by inappropriate thresholding for generalization. However, BLOD3 is the original (most detailed) building model and consists of many minor structures. (This can be observed from the examples presented in previous figures.) Therefore, it is expected to have a significant reduction in terms of point and polygon numbers, but the geometric shapes or characteristics of the models do not deform too dramatically. In addition, the buildings are treated independently, thus building aggregation may seem necessary. Whether to aggregate building groups should depend on the objective of applications. If it is necessary, aggregation should be performed with additional merging process.

5 CONCLUSIONS

This paper presents systematic approaches to create multiple levels of detail for digital terrain and building models. For terrain meshes, a tile-based quad-tree processing with thresholds determined from ground sampling distance under different viewing parameters is suggested to generate terrain LODs. A mesh refinement procedure to correct discontinuities (T-junctions) among tile edges is also presented, which can eliminate discontinuities between adjacent tile meshes effectively and have little impact to the overall rendering performance. For 3D building models, an iterative procedure based on algorithms for generalization on 2D orthographic projections and reconstructing simplified 3D models is proposed. For special structures of building models, additional processes are developed to simplify them but maintain their characteristics. The developed algorithms are effective for individual building models and large building groups with various degree of complexity.

Examples demonstrated in this paper indicate that the developed algorithms can be used for multi-resolution representation of complicated terrain and building models effectively and efficiently. All together, the proposed methods can increase the performance of large cyber city implementation for real-time visualization and applications. More importantly, while reducing the data amount to transmit and process, the proposed multi-resolution representation methods can preserve important geometric and visual characteristics of complicated terrain and building models

ACKNOWLEDGEMENTS

This study was partially supported by the National Science Council of Taiwan under project No. NSC-98-2221-097-MY2.

References

Anders, K., 2005. Level of detail generation of 3d building groups by aggregation and typification. In: Proc. of 22nd International Cartographic Conference, La Coruña, Span.

Blow, J., 2000. Terrain rendering at high levels of detail. In: Proceedings of Game Developers Conference, pp. 903–912.

Chen, L.-C. and Lee, L.-H., 1992. Progressive generation of control frameworks for image registration. Photogrammetric Engineering and Remote Sensing 58(9), pp. 1321–1328.

Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F. and Scopigno, R., 2003. BDAM-batched dynamic adaptive meshes for high performance terrain visualization. In: Proceedings of EUROGRAPHICS, Vol. 22number 3, pp. 505–514.

Coope, I. D., 1993. Circle fitting by linear and nonlinear least squares. Journal of Optimization Theory and Applications 76(2), pp. 381–388.

Forberg, A., 2007. Generalization of 3d building data based on a scale-space approach. ISPRS Journal of Photogrammetry & Remote Sensing 62, pp. 104–111.

Gallier, J., 2000. Curves and Surfaces in Geometric Modeling. Morgan Kaufmann, San Fancisco, CA USA.

Gröger, G., Kolbe, T. H., Czerwinski, A. and Nagel, C., 2008. OpenGIS city geography markup language (CityGML) encoding standard. Technical Report OGC-08-007r1, Open Geospatial Consortium Inc. v. 1.0.0.

Kada, M., 2007. Generalization of 3d building models by cell decomposition and primitive instancing. In: Proc. Joint ISPRS Workshop on Visualization and Exploration of Geospatial Data, Stuttgart, Germany.

Losasso, F. and Hoppe, H., 2004. Geometry clipmaps: Terrain rendering using nested regular grids. ACM Transactions on Graphics 23(3), pp. 769–776.

Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B. and Huebner, R., 2003. Level of Detail for 3D Graphics. Morgan Kaufmann Publishers.

Mayer, H., 1998. Three dimensional generalization of buildings based on scale-spaces. Technical report, Dept. of Photogrammetry and Remote Sensing, Technische Universität München, Germany.

Pouderoux, J. and Marvie, J.-E., 2005. Adaptive streaming and rendering of large terrains using strip masks. In: Proceedings of ACM GRAPHITE 2005, pp. 299–306.

Tsai, F. and Chiu, H.-C., 2008. Adaptive level of detail for terrain rendering in cyber city applications. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII-B4, pp. 579–584.

Tsai, F., Liu, H.-S., Liu, J.-K. and Hsiao, K.-H., 2006. Progressive streaming and rendering of 3d terrain for cyber city visualization. In: Proc. ACRS2006, Ulaanbatar Mongolia.