Feature-Based Simplification of Process Plant Models over Network

Zhiyong Su[1,2], Ming Xia[1,2], Weiqing Li[3], Tao He[1,4] and Weiqing Tang[1,4]

[1]*Institute of Computing Technology, Chinese Academy of Sciences*
[2]*Graduate University of Chinese Academy of Sciences*
[3]*School of Computer Science and Technology, Nanjing University of Science and Technology*
[4]*Beijing Zhongke Fulong Computers Technology Co., LTD*

***Abstract*—Visualization of 3D process plant models is an essential feature to support design review in distributed design environment. The preprocessing time, frame rate and visual fidelity play the same import role in improving the effectiveness and efficiency of the entire design review process. General model simplification algorithms, such as levels of detail (LODs) and hierarchical levels of detail (HLODs), always need unbearable time for preprocessing large-scale models and may distort design feature and make them unrecognizable to the reviewer. To efficiently transmit models, various compression technologies can be applied to shrink the data size. However, transmitting large-scale models over network is still a bottleneck of rendering performance. In this paper, we presented a new, faster feature-based model simplification algorithm to simplify process plant models over network. We first get the model's geometric parameters and topology information from the server before visualization. Then we compute LODs and HLODs according to the model's geometric parameters and composing feature on the client. No triangles are transmitted during visualization. We demonstrate its performance on complex process plant models composed of tens of millions of triangles. Results show that our approach is able to shorten the preprocessing time greatly. And it can achieve considerable speedups in frame rate with little loss in image quality.**

***Index Terms*—simplification, levels of detail, hierarchical levels of detail, process plant, design review.**

I. INTRODUCTION

Process plants, such as refineries and petrochemical plants, are complex facilities mainly consisting of pipelines and equipments. With the faster and higher demands of shortening the design cycle and improving the design quality, Computer Supported Collaborative Work (CSCW) for large-scale process plant projects becomes increasingly important and widely used. Designers from multiple disciplines and different regions create models locally before dispatching them to the server for remote access. In order to ensure design standards, the design reviewers need to walk through and interact with the whole 3D process plant models interactively on the client. The preprocessing time, frame rate and visual fidelity play an important role in improving the effectiveness and efficiency of the entire design review process and shortening the design cycle of a project. As the complexity of the process plant models increases, the enormous size of these CAD datasets poses a number of challenges in terms of interactive display, manipulation and especially preprocessing time on current desktop PCs over the network. By far, although a lot of well-known plant CAD systems have been developed, such as PDS®, PDMS® and SmartPlant3D®, none of the design review tools they provided works well in case of large CAD datasets especially on current desktop PCs.

The common way to visualize a model over the network is to triangulate the model into triangle meshes on the central server and then transmit these meshes to the client for remote visualization [1]–[3]. To alleviate the overload of the network, various model simplification algorithms are applied to shrink the data size. Several acceleration techniques which can reduce the number of rendered polygons have been actively researched for interactive display of large geometric datasets. One is to precompute different LODs of a given object or portions of the model. But most of the earlier algorithms have focused on computing separate LODs of objects in the scene. In order to create higher fidelity and drastic approximations, reference [4] proposed a new approach based on HLODs. In contrast to conventional LODs of objects, it simplifies separate portions of a scene together by grouping and partitioning operations according to the spatial proximity. Although these algorithms can make great contribution to the runtime performance, they may take hours to compute the LODs and HLODs of the models consisting of tens or hundreds of millions of triangles. That's unacceptable for the reviewer in the design review process who thinks that five minutes just seems like an eternity [5][6]. On the other hand, the HLODs may merge objects of different design feature in close proximity and make the simplified object unrecognizable in engineering. For transmission of 3D models over the network, several compression algorithms have been proposed. But they also spend a lot of time to compress the data on the server. And the data should then be decompressed on the client during visualization. Despite of the advantages of the simplification and compression algorithms, the preprocessing time they spent is still unbearable and the complex model is still slow to transmit over the network. To reach a trade-off among the preprocessing time, frame rate and visual fidelity is a potentially better methodology to design review.

Manuscript Received on 30 June 2008.
E-Mail: suzhiyong@ict.ac.cn.

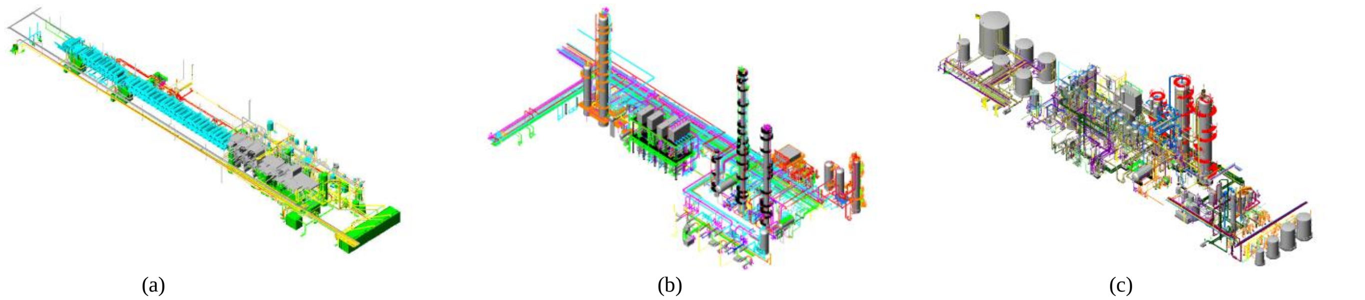


Fig. 1. Three examples of process plant models. (a) contains about 5.7 million triangles and is 213.96×36.05 meters in plan and 19.14 meters in height. (b) contains about 11.6 million triangles and is 149.37×82.96 meters in plan and 65.97 meters in height. (c) contains about 21.6 million triangles and is 119.40×33.66 meters in plan and 31.32 meters in height.

The main goal of this research is to devise an efficient and practical simplification algorithm to enable the interactive display of large process plant models on current desktop PCs over the network. We present a new, faster simplification algorithm in this paper, taking PDSOFT® 3DPiping, a process plant CAD system developed by Beijing Zhongke Fulong Computers Technology Co., LTD (http://www.pdsoft.com.cn), as the background. We only transmit the geometric parameters and topology information to the client before visualization and do the model simplification locally. We compute the LODs of the equipment, pipe and component according to their shape feature and geometric parameters. The simplification of the pipeline is classified into three types according to its distance to the reviewer. The HLODs of the pipeline are generated by simplifying the relative voxels together according to their shape feature and topology. And when the pipeline is far away enough from the reviewer, it would be represented by its smart lines. As compared to previous LODs and HLODs algorithms, our approach can greatly shorten the preprocessing time for computation under the premise of guaranteeing the frame rate and visual fidelity the design reviewers need. And our approach does not need to transmit the triangle mesh before and during the whole visualization.

The rest of this paper is organized as follows. We first survey related work on LODs and HLODs algorithms in Section Ⅱ, and will particularly discuss their problems in handling the process plant models in design review process. Section Ⅲ gives a detailed analysis of the composing feature and topology of the process plant models. Based on the discussion and analysis above, we will then develop and describe our new approach to such models: After giving an overview of our approach in Section Ⅳ, we will then describe the details of our LODs and HLODs generation algorithms in Section Ⅴ. Section Ⅵ then discusses our implementation and its performance. Limitations and future work are discussed in Section Ⅶ. Finally, we summarize our results in Section Ⅷ.

II. RELATED WORK

There is an extensive literature on simplification of large models. In this section, we briefly survey the LODs and HLODs algorithms which have influenced our work and then discuss their problems in simplifying our process plant models for design review.

The concept of LOD was proposed by Clark [7] in 1976. It uses hierarchical representations of models and computes multiple levels of detail to reduce the number of polygons rendered in each frame. The various operators that have been used for simplification of meshes include vertex removal, edge collapse, face collapse, vertex clustering, vertex merging, etc. [8]–[11]. Reference [4] presented a new approach based on the hierarchical use of static LODs and HLODs, which can gain drastically simplification and better fidelity on the whole scene. HLOD is a generalization of the level of detail concept to hierarchical aggregations of objects. In contrast to conventional LODs of objects, HLODs are generated by simplifying separate portions of a scene together to create higher fidelity and drastic approximations. There are three basic types of LOD: discrete, continuous, and view-dependent LOD [12]. The traditional discrete LOD is simple and works best with most current graphics hardware. Continuous LOD creates a data structure encoding a continuous spectrum of detail. The desired level of detail is then extracted from this structure at runtime. It provides better granularity, which in turn can provide better fidelity. View-dependent simplification can provide even better fidelity for a given polygon count, and can handle models (such as terrains) containing very large or poorly segmented individual objects, but it increases the run-time processing and memory requirements. All of these approaches can be useful in different situations. Though continuous and view-dependent simplification are elegant and provide useful capabilities, they increase the run-time processing and memory requirements. Despite advances in continuous and view-dependent LOD, traditional discrete LOD remains by far the most common approach in practice [12]. We also use static LODs and HLODs in our system.

Due to the practical and industrial importance of simplifying complex models, there exists a vast suite of different LOD and HLOD approaches to this problem. However, many of them are not able to effectively and efficiently simplify process plant models for design review due to the following two reasons. One of the reasons is the long preprocessing time spent in LODs and HLODs computation due to their lack of knowledge of model feature. Take the Power Plant model which consists of about 12M triangles as an example, the generation of LODs and HLODs takes nearly 4 hours on the workstation [4][13]. Another reason is that HLODs may merge objects of different design feature in close proximity and make the simplified object unrecognizable. Design review has a strict requirement for the simplification: design feature, such as the layout and topology connection, of the model near to the reviewer must be preserved and displayed accurately. Therefore, it is necessary to develop an effective and efficient model simplification approach for the design reviewer to perform model checking in collaborative process plant CAD.

III. PROCESS PLANT MODEL

As our simplification algorithm is built on the composing feature and topology of the process plant models designed by our PDSOFT® 3DPiping, we first give a detailed analysis of the geometry modeling and topology modeling of process plant models in this section.

3.1 Geometry Modeling

Process plant models are mainly composed of equipments and pipelines. The pipeline includes pipes and components. The traditional research in CAD focuses on fields in mechanical design, such as solid modeling, feature-based modeling, surface intersecting etc. However, in process plant design CAD, designers mainly concentrate on the layout of tremendous number of components under complex constraints. Pipes, various kinds of equipments and components are created by the basic voxels, such as box, cylinder, prism, sphere, etc., by CSG (Constructive Solid Geometry, CSG) methods. Fig. 1 shows three examples of the process plant model. Fig. 2 shows the main voxels used in our PDSOFT® 3DPiping. As the engineering graphic library of the process plant CAD system should provide not only the 3D wire frames and solid, but also the 2D graphics to support the automatic generation of engineering drawings and the component topology expressed by lines and points, we have designed a parametric graphics descriptive language and an engineering graphic library. Such an approach makes it easy to create intelligent information models of pipes, components and equipments with comprehensive geometric, topological and engineering semantics description. The redundancy of the engineering graphic library is reduced by parametrically describing a class of graphics with geometry similarity [14].

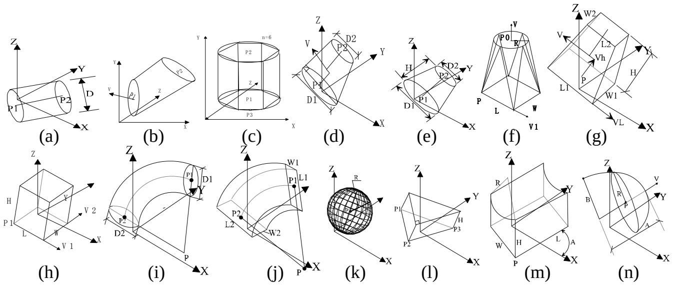


Fig. 2. Main voxels used in our collaborative process plant CAD.

3.2 Topology Modeling

Topological connection is one of the most important research fields in process plant design. We have developed a new approach to describe the topological structure of the process plant models based on the duality point and smart line in our PDSOFT® 3DPiping [15]. The duality point is used to describe the topological adjacency between two piping components, such as pipe segment, tee, valve, equipment and so on. The smart line is a special line of the pipe or component, which reflects topological attribute of the pipes and components. Usually, the smart line is represented by the centerline of the pipeline. A simple example of the duality point and smart line is illustrated in Fig. 3.

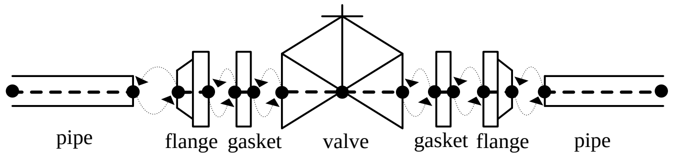


Fig. 3. A simple example of the pipeline. The black point is duality point. The dashed is smart line. Note that a pipeline may include a certain number of pipes and components.

In order to give a detailed analysis of the composing feature, we take the models in Fig. 1 as an example. The statistics of objects, voxels and triangles of the three models are showed in Table 1, Table 2 and Table 3 respectively. We can find out that the total triangles of the pipelines may add up to 80% of the whole model. This means that the simplification of pipelines plays an important role in the rendering performance.

TABLE 1: STATISTICS OF OBJECTS OF THE MODELS IN FIG. 1. THE NUMBER OF PIPELINES IS THE SUM OF PIPES NUMBER AND COMPONENTS NUMBER.

Model	Equipments	Pipes	Components	Pipelines
Fig.1(a)	201	3,491	12,372	841
Fig.1(b)	108	8,509	28,959	1,101
Fig.1(c)	232	7,787	31,059	1,543

TABLE 2: STATISTICS OF VOXLES OF THE MODELS IN FIG. 1.

Model	Equipments	Pipes	Components	Total
Fig.1(a)	7,919	3,491	1,0762	22,172
Fig.1(b)	2,148	8,509	32,751	43,408
Fig.1(c)	9,382	7,787	55,885	73,054

TABLE 3: STATISTICS OF TRIANGLES OF THE MODELS IN FIG. 1.

Model	Equipmens	Pipes	Components	Total
Fig.1(a)	989,577	335,136	4,367,843	5,692,556
Fig.1(b)	336,792	816,864	10,425,388	11,579,044
Fig.1(c)	688,685	747,552	20,183,035	21,619,272

IV. OVERVIEW

In this section, we give a brief overview of the main components of our model simplification approach. These components are the key issues of model simplification for design review, the CAD datasets transmission for design review, the scene representation, the tradeoffs among static, continues and view-dependent LODs approaches and the simplification algorithm for our process plant models.

4.1 Key issues of model simplification for design review

The frame rate and visual fidelity are certainly the main focus of the simplification algorithms. Most of the earlier researches have focused on how to reduce the number of polygons as many as possible and keep the model's visual appeal. The preprocessing time that the simplification algorithms take, by contrast, receives less attention. However, in practical engineering application, it is as important as the frame rate and visual fidelity. Minimizing the preprocessing time spent in transmission and simplification is a priority in design review.

The reviewer mainly concerns the design feature of the model, such as the layout and topology connection, during visualization. Therefore, the portion of the model close to the reviewer needs to be rendered accurately both in pixel and engineering. Any simplification that will mislead the reviewer is forbidden.

4.2 CAD datasets transmission for design review

To alleviate the overload of the network, we only transmit the voxels' geometric parameters and the topology information of the whole model. Engineering information of equipments, pipelines, pipes and components may be accessed via network according to their IDs. The design reviewers firstly choose the model from the server on the client. After receiving the request, the server organizes the voxels' geometric parameters and the model's topology information from the database and then sends them to the client by web service. Fig. 4 shows the data structure of the equipment, pipe, component and pipeline.

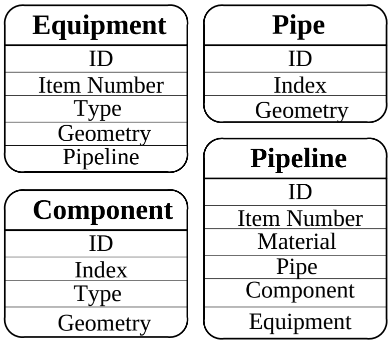


Fig. 4. Data structure of the equipment, pipe, component and pipeline for transmission. Note that the "Geometry" field only contains the geometric parameters of the voxels. Take the sphere as an example, we only transmit its radius and center point rather than its triangle meshes.

4.3 Scene Representation

Our process plant CAD datasets consist of a large number of basic voxels, such as box, cylinder, prism, sphere, etc., which are organized according to a functional, rather than spatial, hierarchy. This kind of organization is not optimized for rendering performance. We first group the voxels according to the functionality, and then subdivide the large pipelines into sub-pipelines. After the grouping and partitioning, we compute the scene graph in a top-down manner on the set of equipments and pipelines by BSP tree. The equipments and pipelines become the leaf nodes in the scene graph.

4.4 Discrete LODs for rendering large process plant models

As mentioned in Section 2, although continuous and view-dependent LODs are elegant and can provide better fidelity, they impose significant memory and processor overhead during visualization. Another issue is that current graphic hardware can render display lists faster than immediate mode primitives. Existing continuous and view-dependent algorithms are inherently immediate mode and, therefore, cannot take advantage of display lists. Given our emphasis on runtime performance, we use static LODs and HLODs, and render them using display lists.

4.5 Model Simplification

We compute the LODs of the equipment, pipe and component based on their voxels' shape feature and geometric parameters. The simplification of the pipeline is classified into three types according to its distance d to the reviewer. Let d_1, d_2 be the distance thresholds, which are proportional to the radius of the pipeline, and $d_1 < d_2$. If the $d < d_1$, the pipeline's LODs are computed directly by triangulating its voxels. The HLODs of the pipeline are generated by simplifying the relative voxels together according to their feature and topology, where $d_1 \leqslant d \leqslant d_2$. If the $d > d_2$, the pipeline is represented by its smart lines.Details of the model simplification will be discussed in Section 5.

V. MODEL SIMPLIFICATION

In this section, we firstly discuss the scene graph of the process plant model. Then we present the details of our model simplification algorithm.

5.1 Grouping and partitioning

Note that all of the operations in design review are carried out based on the object level rather than the voxel level. By "object" we mean the equipment, pipeline, pipe and component. We first group the object's voxels to form the object node by functionality. Fig. 5 shows the structure of object nodes used in our scene graph.

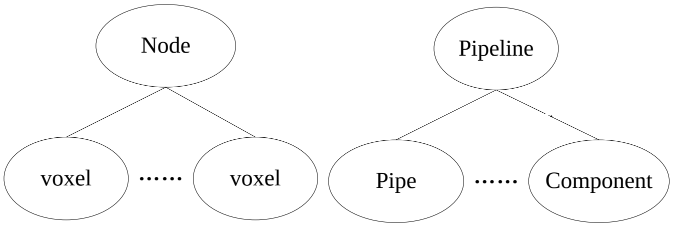


Fig. 5. Structure of object nodes. The "Node" in the figure includes the equipment, pipe and component. The pipe is represented by a cylinder in our system. The pipeline always consists of many pipes and components.

As we use static LODs and HLODs, spatially large objects may pose a problem. When the design reviewer is close to any region of a spatially large object the entire object must be rendered in high detail, even though portions of it may be very far from the reviewer. In process plant models, there are some long pipelines need to be partitioned. We compute an approximate AABB bounding volume for each pipeline based on its smart lines instead of its triangles. If the size (bounding box diagonal) is greater than a specified threshold, we subdivide the pipeline along the longest axis of its bounding box into two pipelines which share the same engineering information. There may be some equipments which occupy large space, such as vessels and furnaces. But their shape is simple and the amount of their voxels is small. Thus, we don't subdivide large equipments into multiple sub-equipments.

After the grouping and partitioning, we compute the scene graph in a top-down manner on the set of equipments and pipelines by BSP tree. The equipments and pipelines become the leaf nodes in the scene graph.

5.2 LODs generation for the equipment, pipe and component

We compute the LODs of equipments, pipes and components based on their voxels' shape feature and geometric parameters. As all the voxels are transmitted using parametric form, we can generate their triangle meshes according to their shape feature and geometric parameters. LODs of the voxels could be computed directly by using different triangulation precisions. Each voxel owns a member variable m precision and a member function triangulation(). Every time when triangulation() is called, the voxel is triangulated using the precision m_precision and after that the variable m_precision minus Δ p for the next triangulation. The initial precision and Δ p are proportional to the size of the voxel. We also use the triangulation precision to control the simplification error on the basis of the voxels' shape feature. All the triangle meshes are stored using triangle strips and rendered by display lists.

5.3 Pipeline simplification

As shown in Table 3, the total triangles of pipelines (include pipes and components) may add up to 80% of the whole process plant model. Thus, the pipeline's simplification algorithm plays an important role in improving rendering performance. According to its distance d to the reviewer, we classify the simplification of the pipeline into three different types considering the requirements of design review: $d < d_1$, $d_1 \leqslant d \leqslant d_2$ and $d > d_2$. Fig. 6 gives a simple example of our pipeline simplification algorithm.

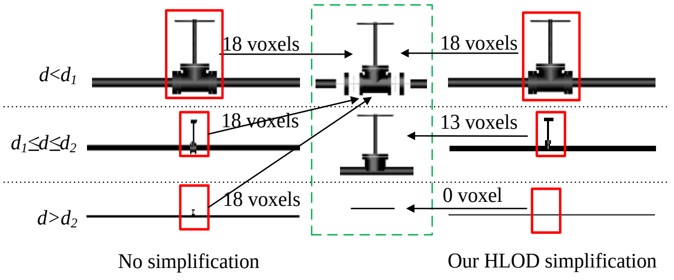


Fig. 6. A simple example of our HLOD simplification. No voxel is merged when d < d1. We merge the relative voxels according to their shape feature and topology when $d_1 \leq d \leq d_2$ and represent the pipeline by its smart lines when $d > d_2$.

1) $d < d_1$: When the reviewer is close to the pipeline, they mainly put their concern in the layout of the pipes and their adjoining components. The design feature should be preserved during visualization and the pipeline should be displayed accurately. Therefore, the LODs of the pipeline are computed by combining the LODs of its pipes and components. None of the voxels is merged under this situation.

2) $d_1 \leqslant d \leqslant d_2$: When the viewpoint of the reviewer is far from the pipeline ($d \geq d_1$), the most important measure of fidelity is not geometric but perceptual in our application. We generate the HLODs of the pipeline by simplifying the relative voxels together according to their shape feature and topology. This will reduce the number of voxels of the pipeline. A detailed description of the HLODs algorithm is given as follows:

- **Step1**: For each pipeline, set P = PipeVec[0], push the components of P in the vector componentVec and travel the next pipe PipeVec[i] (1≤ i < pipeNum) according to the pipeline's flow direction.
- **Step2**: If the smart lines of P and PipeVec[i] are parallel, then execute step 3, otherwise, execute step 4.
- **Step3**: Extrude the cylinder of P to the end of PipeVec[i] to form a new cylinder, which has the same attribute of P. Push the new cylinder in the vector GeometryVec and the components of PipeVec[i] in the vector componentVec. Set P = PipeVec[i], push the components of P in the vector componentVec and then travel the next pipe and execute step 2. If all the pipes are traveled, then go to step 5.
- **Step4**: Push the cylinder of P in the vector GeometryVec. For each voxel of the componentVec, if its type is not among (a)-(e) in Fig. 2, or it's one of them, but none of its duality point is on its pipe's smart line, then we push it in the vector GeometryVec.
- **Step5**: Compute the pipeline's HLODs by triangulating all the voxels in GeometryVec using different precisions.

3) $d > d_2$: The $d > d_2$ means that the pipeline is far enough from the reviewer and is no longer the focus. Under the premise of guaranteeing the visual fidelity, we represent the pipeline using its smart lines. This makes us to gain drastically simplification with little loss in image quality.

VI. IMPLEMENTATION AND PERFORMANCE

We implemented our system using C++ and OpenGL on a dual 2.0 GHz PC, with 4GB of RAM and a GeForce 7200GS with 256MB of video memory. Our server is a PIV 3.0GHz PC with 2GB of RAM. All PCs are connected via 100 Mb/s Ethernet LAN.

We have tested our approach on the three process plant models shown in Fig. 1. These models are designed using PDSOFT® 3DPiping and stored on the central server. We firstly select the model from the model list on the server. Then the server gathers the model's geometric parameters and topology information and sends them to the client by web service. Our model simplification is carried out completely on the client. We compute four LODs for each equipment and two LODs when $d < d_1$, two HLODs when $d_1 \leqslant d \leqslant d_2$ for each pipeline. All the voxels are represented by triangle strip and all the LODs and HLODs are represented using display lists. Only the engineering information is transmitted over the network during the design review process.

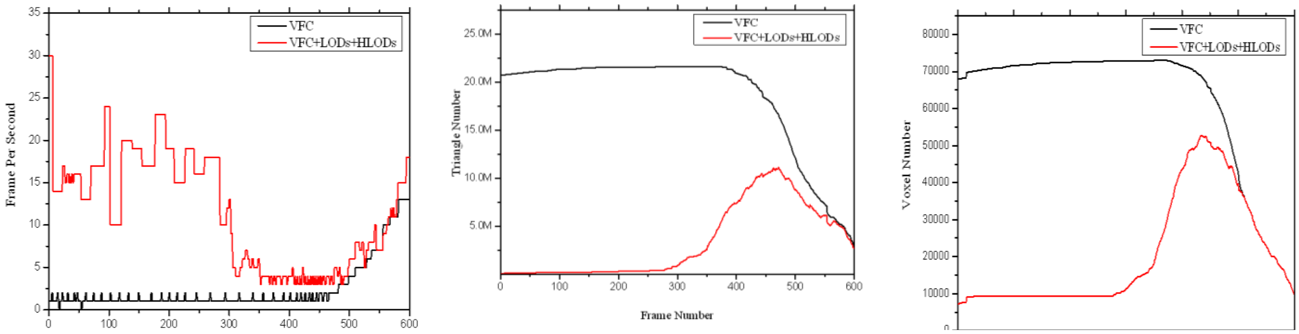


Fig. 7. Comparisons of the frame rate, triangle number and voxel number by rendering the model in Fig. 1(c).

6.1 Preprocessing time

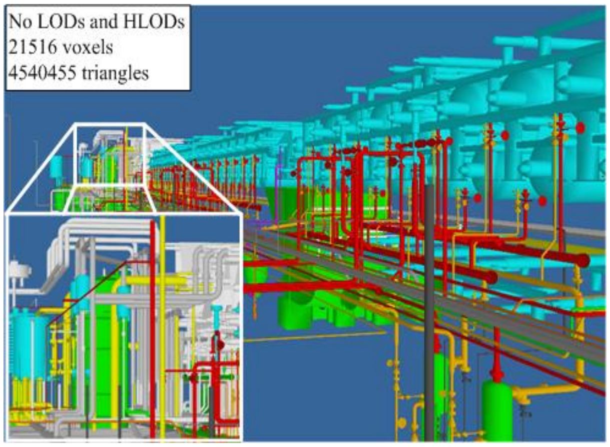
The preprocessing includes both the querying on the server and the LODs and HLODs generation on the client. Table 4 shows the amount of time it took to preprocess the three models. The objects include equipments, pipes and components. We also export these models in XML format as parametric form with topology information organized by the server. The size of these XML files is shown in Table 4 too. The transmission takes at most a few seconds, depending on the model's complexity which is measured by the number of its pipelines in engineering. As our LODs and HLODs are computed locally, no time is needed for data compression and decompression.

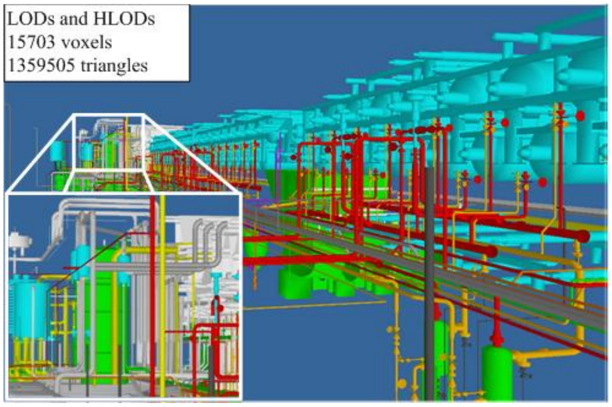
TABLE 4: PREPROCESSING TIME FOR THE QUERYING AND SIMPLIFICATION

Model	Triangles	File Size	Querying	LODs & HLODs
Fig.1(a)	5,692,556	11.29 M	76 secs	20 secs
Fig.1(b)	11,579,044	24.15 M	178 secs	37 secs
Fig.1(c)	21,619,272	31.41 M	195 secs	68 secs

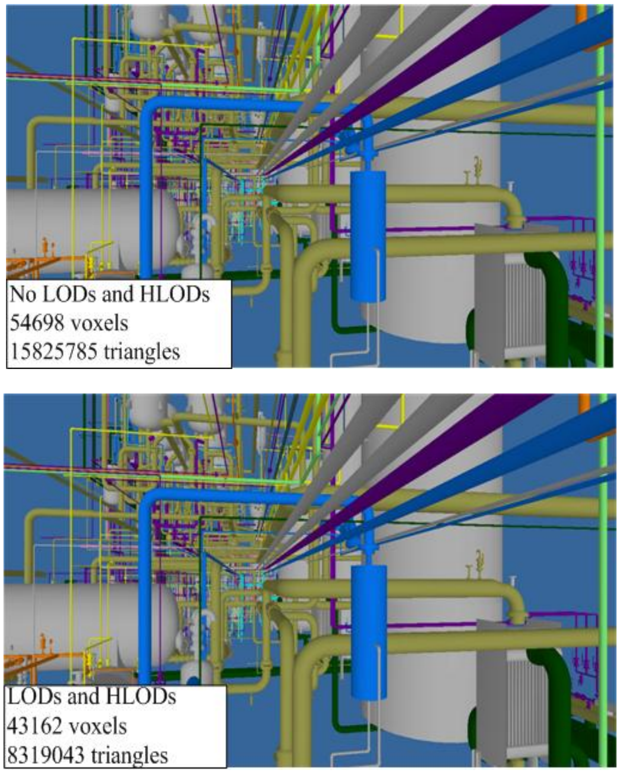
6.2 Rendering performance

Fig. 7 illustrates the performance of our system by rendering the model in Fig. 1(c) over the same path. The graphs compare the frame rate, triangle number and voxel number for using VFC alone and for the combination of VFC, LODs and HLODs. We start with the full model on screen and then navigate into the model randomly.


No LODs and HLODs
21516 voxels
4540455 triangles


LODs and HLODs
15703 voxels
1359505 triangles

(a) A close view of the model in Figure 1(a)


No LODs and HLODs
54698 voxels
15825785 triangles

LODs and HLODs
43162 voxels
8319043 triangles

(b) A close view of the model in Figure 1(c)

Fig. 8. Visual comparison by rendering the model in Fig. 1(a) and Fig. 1(c). The numbers of voxels and triangles of different approaches are also shown in the figure.

6.3 Visual Comparison

We show the visual difference between the model in Fig. 1(a) and Fig. 1(c) rendered with and without our approach in Fig. 8. By merging the related voxels of several objects in the same pipeline and representing pipelines by their smart lines when they are far away from the reviewer, we are able to gain drastic simplification with little loss in image quality.

VII. LIMITATIONS AND FUTURE WORK

Our work is dedicated to develop an efficient and practical simplification algorithm to enable the interactive display and manipulation of large process plant models for the design review on current desktop PCs over the network. As our approach is based on the model feature, it's merely suitable for the process plant models of our PDSOFT® 3DPiping now.

Since we use static LODs and HLODs, our approach inevitably has its limitation in terms of potential "popping" artifacts as we switch between different simplification levels. However, the use of the simplification is mainly to speed up rendering for design review. For this purpose, some inevitable "popping" artifacts are acceptable.

Our system renders an appropriate LOD for every equipment and the pipe and component which are close to the reviewer. Since a HLOD of a pipeline in the scene graph is an approximation for its pipes and components, if we render a pipeline's HLOD in traversing the scene graph, we do not need to visit its descendants. But the time spent in computing the distance and selecting the appropriate LODs to display at runtime is still a performance bottleneck. Hundreds or even thousands of distance computation and querying are executed per frame in average.

Visibility culling is currently the most effective technique for improving rendering performance in complex 3D environments. However, only view-frustum culling is used in our system now. Due to the high depth complexity in process plant models, an efficient occlusion culling algorithm needs to be developed urgently.

Finally, the memory overhead of our approach can be high. In the current implementation, all LODs and HLODs precomputed are stored in the main memory. It would be useful to develop an out-of-core rendering system that uses a finite memory footprint and uses prefetching techniques to load the visible nodes.

VIII. CONCLUSION

In this paper, an efficient and effective model simplification approach for visualization of process plant models over the network in a collaborative process plant CAD is reported. We just transmit the model's geometric parameters and topology information to the client before visualization. Only engineering information queried by the reviewer is transmitted during the design review process. We take the actual requirement of the design review into account and perform different pipeline simplification strategies according to its distance to the reviewer. We compute the HLODs of the pipeline by merging the relative voxels together according to their feature and topology and use its smart lines to represent the pipeline when it's far away enough from the reviewer. We have demonstrated the performance of our system on three complex process plant models designed using our PDSOFT® 3DPiping. The results show that our approach can shorten the preprocessing time spent both in transmission and simplification greatly and improve the rendering performance considerably with little loss in image quality.

ACKNOWLEDGEMENT

We thank the members of PDSOFT® team at Beijing Zhongke Fulong Computers Technology Co., LTD. We also thank the anonymous reviewers.

REFERENCES

[1] J. Y. H. Fuh and W. Li. Advances in collaborative cad: the state-of-the art, *Computer-Aided Design*, vol. 37, no. 5, pp. 571–581, 2005.

[2] Z. M. Qiu, Y. S. Wong, J. Y. H. Fuh, Y. P. Chen, Z. D. Zhou, W. D. Li and Y. Q. Lu. Geometric model simplification for distributed cad, *Computer-Aided Design*, vol. 36, no. 9, pp. 809–819, 2004.

[3] Z. Zheng, P. Edmond and T. Chan. Interactive view-dependent rendering over networks, *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 576–589, 2008.

[4] C. Erikson, D. Manocha and V. B. I. William. HLODs for faster display of large static and dynamic environments. in *Proceedings of the 2001 symposium on Interactive 3D graphics*, Los Angeles, pp. 111–120, 2001.

[5] A. Dietrich, A. Stephens and I. Wald. Exploring a boeing 777: Ray tracing large-scale cad data, *IEEE Computer Graphics and Applications*, vol. 27, no. 6, pp. 36–46, 2007.

[6] D. Kasik, D. Manocha, A. Stephens, B. Bruderlin, P. Slusallek, A. Dietrich, E. Gobbetti, F. Marton, W. Correa and I. Quilez. State of the art in massive model visualization, in *ACM SIGGRAPH 2007* Course #4 Notes, 2007.

[7] J. H., Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, vol. 19, no. 10, pp. 547–554, 1976.

[8] J. Rossignac and P. Borrel. Multi-resolution 3d approximations for rendering complex scenes, in *Geometric Modeling in Computer Graphics*, Springer Verlag, pp. 455–465, 1993.

[9] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, W. Wright. Simplification envelopes, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New Orleans, pp. 119–128, 1996.

[10] H. Hoppe. Progressive meshes, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New Orleans, pp. 99–108, 1996.

[11] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics, in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, Los Angeles, pp. 209–216, 1997.

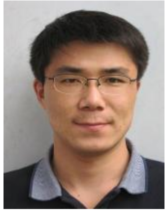
[12] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney and Benjamin. in *Level of Detail for 3D Graphics*, Morgan Kaufmann, pp. 9–13, 2002.

[13] V. B. I. William, A. Sud, N. K. Govindaraju and D. Manocha. Gigawalk: interactive walkthrough of complex environments, in *Proceedings of the 13th Eurographics workshop on Rendering*, Pisa, pp. 203–214, 2002.

[14] X. J. Huang. Research on Fast Rendering Techniques in Engineering CAD. PhD thesis, Institute of Computing Technology, Chinese Academy of Science, 2002.

[15] X. F. Dai. The research on AEC CAD modeling technology based on enhanced graph and multi-state model. PhD thesis, Institute of Computing Technology, Chinese Academy of Science, 2000.

Zhiyong Su is a PhD student in Institute of Computing Technology, Chinese Academy of Sciences. He received his Bachelor's and Master's degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology in 2004 and 2006, respectively. His research interests include real-time rendering, graphics hardware and virtual reality.



Ming Xia is a PhD student in Institute of Computing Technology, Chinese Academy of Sciences. He received his Bachelor's degree from the School of Computer and Information Engineering, Henan university in 2004 and Master's degree from the School of Automation, Nanjing University of Science and Technology in 2007, respectively. His main research interests is Manufacture Information Engineering.

Weiqing Li is an associate professor in School of Computer Science and Technology, Nanjing University of Science and Technology. He received his Ph.D. and Bachelor's degree from the School of Computer Science and Technology, Nanjing University of Science and Technology in 2006 and 1997, respectively. His main research interests are computer graphics, virtual reality and system simulation.

Tao He is an associate professor in Institute of Computing Technology, Chinese Academy of Sciences. He is the CEO of Beijing Zhongke Fulong Computers Technology Co., LTD. He received his Bachelor's and Master's degrees from the School of Computer Science, Wuhan University in 1992 and 1995, respectively. And then he received his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences in 1999. His main research interests are CAD&CG and CSCW.

Weiqing Tang is a professor in Institute of Computing Technology, Chinese Academy of Sciences. He is also the chairman of the board of Beijing Zhongke Fulong Computers Technology Co., LTD. He received his Bachelor's and Master's degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology in 1984 and 1987, respectively. And then he received his Ph.D. from Institute of Computing Technology, Chinese Academy of Sciences in 1993. His research interests include CAD&CG, CSCW and scientific visualization.