Incremental Transmission of B-Rep Models through the Network

Youngjae Song[1] and Kunwoo Lee[2]

[1]Seoul National University, songyj@cad.snu.ac.kr
[2] Seoul National University, kunwoo@snu.ac.kr

ABSTRACT

The concurrent engineering approach is getting more popular in the product development process, and thus it becomes crucial to effectively share the shape information between the various people and the systems in a distributed design environment. There have been many previous researches to efficiently transmit and handle the geometric data, but still the transmission of geometric data is a time-consuming task and results in the inevitable latency.
We introduce the level-of-detail concept into the transmission of boundary representation (B-rep) model to cut down the latency. The original model is broken down into small fragments in the level-of-detail generation step, and the transmission sequence of the data fragments is efficiently scheduled according to their contribution to the shape details. This approach is very attractive when transmitting very complex parts and assemblies, because the user can capture the overall shape at the earlier stage of the transmission.
In this paper, a new level-of-detail generation method for B-rep models, the scheduling algorithm to determine the transmission sequence of data fragments, and the reconstruction method of original model from the data fragments are discussed. Then, the overall system architecture is explored. Finally, some examples are demonstrated to verify the effectiveness of the proposed approach.

Keywords: streaming, LOD, level of detail, multi-resolution, B-rep, solid model

1. INTRODUCTION

The concurrent engineering approach is now being widely accepted in almost every step through the product design cycle. And, the great advance in the network technology enables the seamless communication among the spatially distributed people. Many kinds of people, applications and systems are simultaneously involved in the product design.
A key issue for the collaborative design activity over the network is the effective communication to share the information between the people and the application programs. Many kinds of information and data are sent, received and shared through the product development cycle. Of all this information, the shape information is most difficult to be transmitted through the network. There are two important reasons; (a) the size of data file containing the shape information is generally quite larger than that of the other information and, (b) the shape model data are continuously modified and updated until the final design is finally fixed.

Due to the large size of model data, there is a significant delay during the transmission over the network. One can have two kinds of strategy to solve this problem:
(1) Minimize the size of the geometry data; we might compress the geometry data and change them into more compact form.
(2) Cut down the latency for the data transmission; if we can stream the geometry data, the client user can see the partial shape even during the transmission. If the model has been changed after the transmission, sending just the partial data (modified portion) will be enough to update the client-side model.

The above two strategies can be used simultaneously, because they can be applied independently.
In this paper, the latter approach has been explored. The main contribution of this paper is a new methodology for streaming CAD model. To stream the geometry data, the whole data should be fragmented into smaller data fragments, and also the sequence to transmit the fragments should be determined. In the presented work, we adopt a level-of-detail (LOD) concept during the generation of these data fragments.

So, the transmission sequence of the data fragments can be optimally determined according to their contribution to overall shape of the part. In this way, the user can capture the rough shape of the part from the beginning of the transmission, and get the more detailed shape as the transmission process goes on.

The rest of the paper is structured as follows. Section 2 describes the previous work related to this paper. The overview of the proposed approach is described in Section 3. In Section 4, 5, and 6, the detailed procedures are explored. Also, the result and some examples are presented. Finally, the discussion and closing remark are followed.

2. LITERATURE REVIEW

2.1 Methodologies for geometric data transfer through the network

The methodology for geometric data transfer greatly depends on its representation method of geometry. Generally, many kinds of representations are used to describe the model shape. But, following three representations are most widely used to transmit the shape data over the network: (a) facet-based representation, (b) feature-based representation, (c) boundary representation of solid model[1].

In the facet-based representation, the entire model is described in the form of a set of polygons. Some further information (such as feature tag, color, etc.) can be added to each polygon or a group of polygons. This representation is very useful to visualize a model, because the common graphics hardware pipeline directly accepts this data to render the resulting shaded image on the screen.

This approach has an advantage of requiring less computing power on the client computer. Most heavy computations are processed in the server-side, so the client needs only to visualize the received data. Also, many fast and efficient algorithms are available to transmit the facet data. Taubin, et al proposed the platform-independent transmission methods[2]. Kim, et al and Lee, et al proposed the method to incrementally update and transmit the facet-based model[3][4]. The disadvantages of this approach are that it is impossible to describe the exact shape of a smooth surface, and that the accuracy depends on the total number of polygons. To express the smooth shaped model to a detail resolution, quite many polygons are required. So the total data size to be transmitted is greatly increased as the accuracy is increased.

The feature-based representation is a parametric, history-based representation. It provides more intuitive and functional information involved in the modeling phase, so it is used as a high-level representation in the most commercial CAD systems. The model feature information includes all the information to exactly describe the shape of model, and also is very compact compared to other representations. So, the shape information and model changes can be transmitted very concisely. The disadvantage of this approach is that the feature information is just descriptive information so it still needs another representation (for example, the boundary representation) to evaluate/display the shape. And, also because the heavy computations (such as to generate the visualization model, or the boundary-representation model) should be performed in the client side, the client computer should be more powerful[5]. Moreover, there are some severe compatibility problems between applications, because each application may interpret the feature model differently depending on the consideration. For example, a protrusion feature in the design viewpoint is most likely interpreted as a step-feature in the manufacturing view. Thus, the process of feature mapping and feature recognition should be performed to overcome this problem[1].

The boundary representation (B-rep) is most widely accepted form in the current commercial computer-aided design systems. The size of B-rep model is larger than the feature-based representation, but still quite concise compared to that of the facet-based representation. Also, B-rep model can precisely describe the geometry of a model.

Di Wu, et al proposed the “streaming” method of solid model, so that geometry model can be stored, retrieved or transmitted in an incremental form and reconstructed on-the-fly[1]. To stream the manifold solids, topology and geometry data are encoded using the one-dimensional dynamic stack structure, and the encoded data is transmitted to client side and decoded into the original solid geometry. Gadh and Sonthi proposed to use the different level of geometric shape abstraction to reduce the amount of data transfer based on given end application. They classified the geometry data into 6 layers and assigned the access level according to the requirements for the given end applications[6].

In this paper, only the B-rep model is taken into consideration. The feature-based representation is most concise, but even in feature-based representation the B-rep model is still required to evaluate the shape of the model. And the time to transfer the feature information is quite trivial compared to the time to generate the B-rep model. Most of all, the feature information contains all the design information and accompanies a risk of exposing design know-hows to other companies.

Facet-based representation is too big to handle in network transmission to express the model precisely. The sub-division scheme can be considered as the alternative, but there is still no merit when compared with the B-rep model itself. And, the B-rep is actually the base data structure of the most concurrent commercial CAD systems, so it is also most acceptable one for the easy integration with them.

Considering the above facts, the B-rep model is most appropriate when sharing the CAD models between the spatially distributed designers and engineers.

Lately, dynamic segmentation and incremental editing algorithm of B-rep model has been proposed by Di Wu, et al[7]. Using the application interface specification (AIS) approach, the proposed algorithm can be used architecture-independently. They used the cellular structure called as "cellular change model(CCM)" to describe the incremental change in the shape. These CCMs are detected and segmented in application-server-side, and then are transmitted and embedded into the current model in the application-client-side to build the final changed model. Also, Di Wu and Radha Sarma extended their work to be able to handle the dynamic and incremental editing of B-rep for collaborative environment[1]. Jie Li, et al suggested a direct incremental transmission of B-rep model[8]. In their research, the incremental model is created for each modeling change and this incremental model is transferred to the client side from the server to update the model. But, to create the incremental model, the whole B-rep structure should be traversed and compared between the previous model and the current model for each modeling stage. So, the algorithm may require a severely time-consuming process for the very complex model.

These researches are closely related to our research. But, in their researches the transmission sequence of the data fragments is not considered for the network latency.

In our approach, the sequence of the transmission is more effective than their approaches because the client always can see the overall shape at the earlier stage and get more detailed shape as the transmission is proceeded. So, the total delay time for user to get the necessary shape information from the geometry database can be greatly shortened.

3. OVERVIEW OF THE PROPOSED APPROACH

3.1 A framework for streaming the B-rep model

We can assume that the transmission problem of the B-rep model can be modeled as the data transmission between two different applications which are running on the different computers. Each computer is connected over the network. When the "Application B"(client side application) requests a solid model from "Application A"(server side application), the target solid model should be transmitted through the network. In this paper, only the transmission problem is discussed, and the data translation will not be discussed. So, we can assume that both "Application A" and "Application B" is using the B-rep model as their common data representation and there is no data translation required between "Application A" and "Application B".

When a B-rep model is transmitted from one to the other, the following processes are executed sequentially.

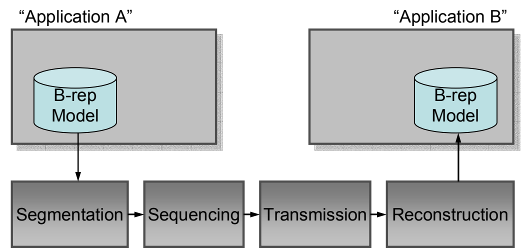

Fig. 1. Framework for streaming B-rep data

(1) Segmentation: the B-rep data of the product model is too big to be transmitted at a time, so it should be decomposed into smaller pieces of data.

(2) Sequencing: the transmission order to send data to the client side should be determined before the transmission process begins.

(3) Transmission: at this stage, each data fragment is transmitted from the server side application to the client side application.

(4) Reconstruction: when the client application has more than two data fragments, the reconstruction process can be executed without stopping the data receiving process. As the transmission goes on, the target model is gradually reconstructed from the received data fragments.

The overall transmission process is shown in Fig. 1.

There are many possible ways to decompose the B-rep model into data fragments in the segmentation stage. Among many possible ways, the LOD concept is adopted in this paper. Our aim in the proposed approach is to decrease the time to present the model in the display. Also, it is desired for the user to see the most brief shape of the target model (lowest level of detail) when the user sees the initial reconstructed model. As the transmission goes on, more and more detailed shape (higher level of detail) is desired to appear. This enables the user to be able to catch the overall shape of the model at the very early stage of the transmission. To achieve this goal, we use the wrap-around algorithm as follows.

3.2 Wrap-around algorithm for Level-Of-Detail model generation

In this paper, we use the wrap-around algorithm that is being developed in the CAD Laboratory, Seoul National University (in Korea).

Originally, the wrap-around algorithm is developed to generate the multi-resolution model from the B-rep model. Unlike many other LOD algorithms, the wrap-around algorithm is applied to the B-rep model, not to the polygonal model. The algorithm can be used not only for a part but also for an assembly model with many sub-components. In this paper, we use the part-level wrap-around algorithm only, so the part-level algorithm will be briefly described. Please, refer to the reference [9] for the more complete explanation.

At first, the algorithm searches the local face-sets consisting of a single face or inter-connected faces which can be deleted to get the simpler shape. If a face-set satisfies a certain requirements, it can be removed from the model and the simpler model can be reconstructed by extending and trimming the neighborhood faces of that face-set. These face-sets are called the 'suppressible features'. Following five kinds of suppressible features are handled in wrap-around algorithm in the current implementation

- Concave features(concave face-sets)
- Hole features
- Chamfer features
- Fillet/round features
- Some kinds of the complex features

The more complicated and intensive algorithms are used for detecting features, the more various kinds of suppressible features can be handled. But too much time-consuming algorithm for detecting features is not desirable for the dynamically changing models, so simpler and faster algorithm requiring less computation time compared to the general feature recognition algorithm is used in the wrap-around algorithm.

After all suppressible features are detected, the features are re-arranged to the increasing order of its volume-contribution ratio, or equivalently of its area. The smallest and most negligible feature is removed at first according to this order. Then, it generates a more simplified model by sequentially removing the suppressible features. Schematic flow of part simplification process using wrap-around algorithm is illustrated in Fig. 2 for a simple example.

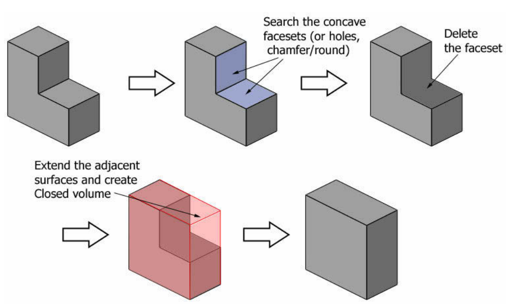

Fig. 2. Schematic flow of part simplification process in wrap-around algorithm

One of the important features of wrap-around algorithm is the reversibility of the simplification process. During the simplification process the removed face-sets are stored as the sheet bodies, and these sheet bodies can be used to restore the un-simplified model from the simplified model. Thus, through the simplification process, the original detailed part can be eventually decomposed into a lowest level-of-detail solid body (the most simplified shape) and many sheet bodies. When desired, the lowest level-of-detail solid body and many sheet bodies can re-construct the original part.

Because of this reversible feature of wrap-around algorithm, the simplification process of the algorithm is called as wrap-around forward operation, and the reverse process (re-constructing process) is called as wrap-around reverse operation.

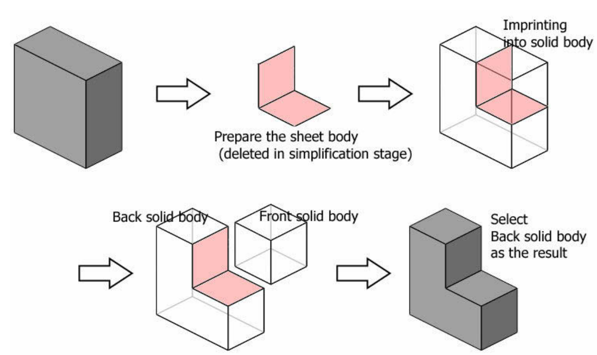

Fig. 3. Schematic flow of the reverse operation of part simplification

3.3 Applying wrap-around algorithm in transmission of B-rep model

Using the wrap-around algorithm, we can decompose the complex solid body into the several geometric features and a simplest solid body. This decomposed information has a great advantage when used in the network transmission. The transmission sequence of the data-fragments can be determined very easily and efficiently using the LOD concept. The simplest solid body is transferred firstly, and the deleted features (stored in the form of sheet bodies) are transferred sequentially according to their contribution to the final shape. The user at the receiving side can capture the overall shape of the body at the early stage of the transmission. So, the total response time for browsing the model in a networked database can be severely shortened.

The reversibility of simplification process plays a key role in the proposed transmission algorithm. The wrap-around forward operation is used in the segmentation stage of the proposed approach. Then, the decomposed

resulting bodies (consisting of one solid body and many sheet bodies) are transmitted sequentially to the client side through the network in the transmission stage. Finally, the client side application uses the wrap-around reverse operation to re-construct the original part from the decomposed bodies.
Each step of the total transmission process will be described with some examples in the following sections.

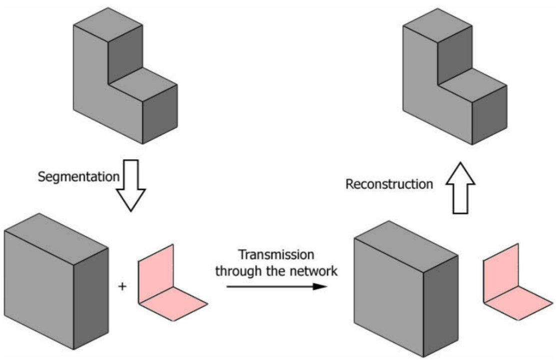

Fig. 4. Applying wrap-around algorithm in transmission of B-rep through network

4. SEGMENTATION

Level-of-detail Generation Step

The process in the segmentation stage is quite straight-forward. The only thing required in this stage is just to apply the wrap-around forward operation to the target part, until its shape can not be simplified further. During the simplification process, all the deleted face-sets are converted and saved in the sheet body form.
Finally, the target part is decomposed into a most simplified solid object and a series of the sheet bodies after the segmentation process.

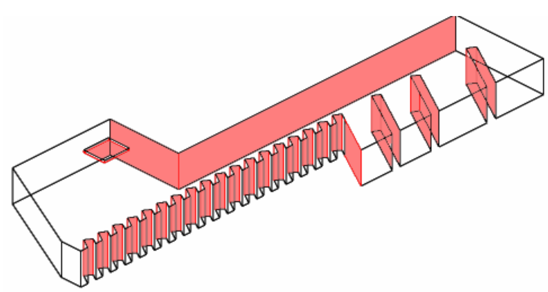

Fig. 5. Segmentation result for an example part

5. SEQUENCING

The first object to be sent should be the solid body resulting from the simplification (most simplified one). But for the transmission sequence of the remaining sheet bodies, there are many kinds of sequencing methods depending on the user's request.
One simple strategy is to sort the sheet bodies in the descending order of their areas.
However, it is desired to preserve the hierarchy among the sheet bodies intended by the designer during simplification. Specifically, when there are repeating patterns of geometry in the target part, the client user may need to see that geometry pattern as a whole. An example of this case is shown Fig. 6. A user may want to see the gear teeth as a whole instead of treating each gear teeth as a separate entity. Or, some other users may want to show several gear teeth and to skip the transmission of the other gear teeth to shorten the delay. These kinds of the sequencing options can be chosen by the users.

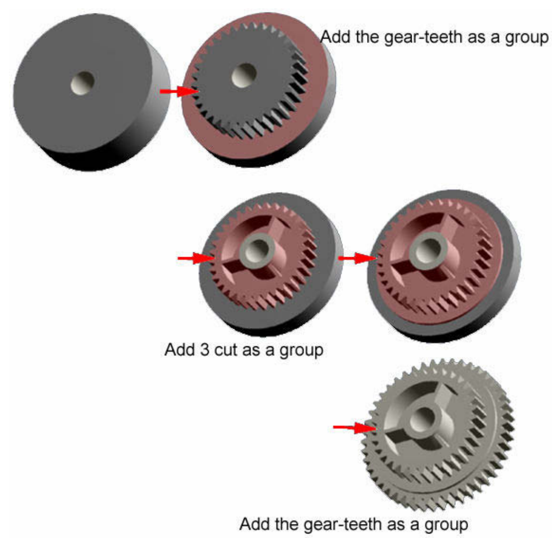

Fig. 6. An example part with several geometry patterns

But, it is very complex and time-consuming to exactly recognize the geometry pattern. So, we use a simple area-based grouping method, that is, we convert the group of the sheet bodies having the same surface area values into a single node in the hierarchical structure. This method may look too simple to identify the geometry pattern, but is really fast and good enough for many examples in the real world. Absolutely, more strict and complex method can be used if a reasonable solution in terms of computational efficiency becomes available.
The size of a data-fragment per transfer is also an important parameter for the transmission efficiency. For a relatively simple model one-feature-per-each-transfer strategy works quite well. But there may be too much small features to be transferred for a very complex model.

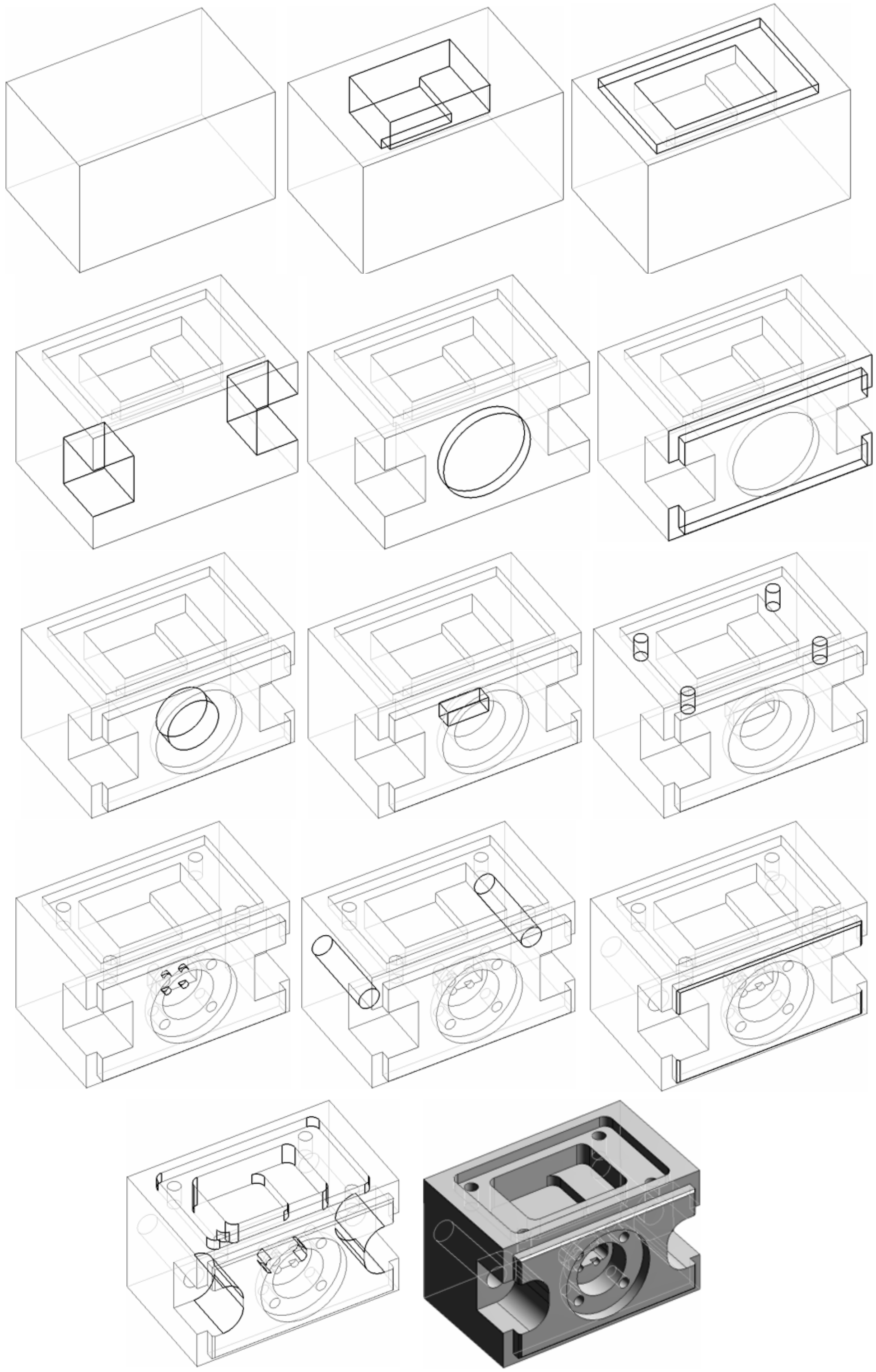

Fig. 7. Reconstruction result of target part

For browsing a large and complex model database in the high-speed internet environments, several successive features can be bundled up to form a lager data-fragment.

6. RECONSTRUCTION

The reconstruction process begins after the transmission of the simplified solid body and the first sheet body is completed. This process runs simultaneously during the transmission task between the server and client is carried out, so the original shape is reconstructed almost just after the transmission ends. The reconstruction result for an example part is shown in Fig. 7. And, the screen shot of server and client program is shown in Fig. 8.

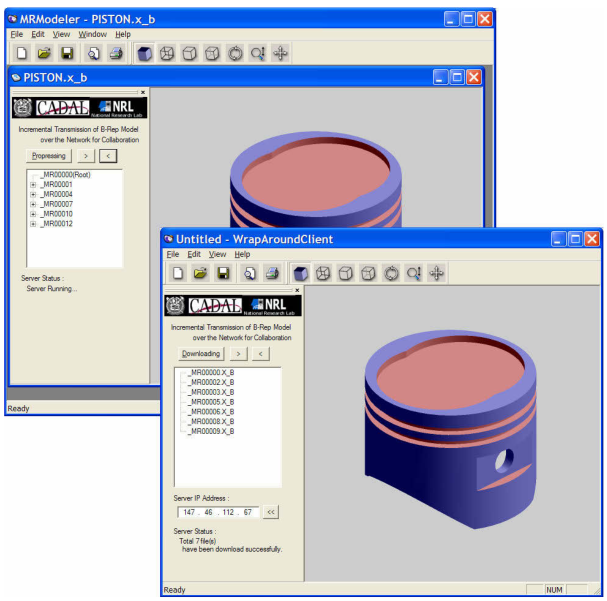

Fig. 8. Screen shot of server and client

7. RESULTS AND DISCUSSION

A new streaming algorithm for B-rep is proposed and implemented. The Level-of-Detail concept is adopted to determine the transmission sequence. The users at the client side can capture the overall shape of the part from the early stage of the transmission and the more and more detailed shape as the transmission goes on. So, the users can inspect the shape freely in the middle of the data transmission.

The new approach is based on the wrap-around algorithm originally developed to generate the LOD of B-rep model. By virtue of the reversible feature of the wrap-around, the algorithm can be applicable to the streaming of the B-rep model.

7.1 Complexity Analysis

The algorithm complexity of the propose approach is almost the same as the complexity of the wrap-around algorithm.

For wrap-around forward operation, the complexity of detecting the suppressible features and that of removing the suppressible features depend on the number of the faces in the B-rep. For both of them, the complexity of the operation is O(n). (n is the number of faces in the B-rep.) And the operations are executed sequentially, so the total complexity of the wrap-around forward operation is also O(n).

For wrap-around reverse operation, the sectioning algorithm is used to divide the body into two pieces in the re-constructing process. This sectioning algorithm also depends on the number of the faces in the model, O(n).

So, the total complexity of the proposed algorithm is O(n).

7.2 Limitations of the algorithm

There are several short-comings in the current implementation due to the limitations of wrap-around. Thus the proposed algorithm in the current implementation can not handle all kind of geometry shape.

Since wrap-around is still in the development process, it is expected that many of the problems will be solved and the proposed approach will be applied efficiently for various kinds of the parts. At the beginning of this research, the wrap-around algorithm can not handle the complex shape features such as multi-face hole loop. But, now it handle the many types of the multi-faced hole loops in Fig. 9.

There are still some unresolved complex features for wrap-around algorithm. But, the proposed approach can be applied to the parts including the unsupported shapes, because these unsupported shapes just remain in the most simplified level of the part. So, the proposed approach can be applied to all kinds of models.

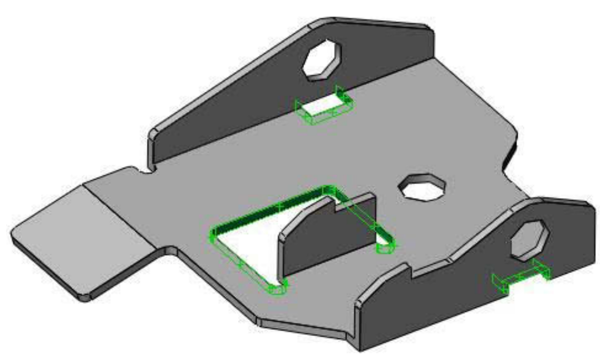

Fig. 9. Screen Complex shape features (multi-face hole loop)

8. ACKNOWLEDGEMENTS

The research is funded by the National Research Laboratory Program (2000-N-NL-01-C-082) of the Ministry of Science and Technology in Korea.

9. REFERENCES

[1] Di Wu and Radha Sarma, Dynamic Segmentation and Incremental Editing of Boundary Representations in a Collaborative Design Environment, Proceedings of Sixth ACM Symposium on Solid Modeling and Application, 2001, pp.289-298

[2] G. Taubin, and J. Rossignac, Geometric Compression Through Topological Surgery, ACM Transactions on Graphics, Vol. 17, No.2, 1998, pp.84-155

[3] H. Kim, J. Y. Lee, and S. B. Han, Process-Centric Distributed Collaborative Design Based on the Web, Proceedings of the 1999 ASME-DETC Computers in Engineering Conference, Las Vegas, Nevada, September, 12-15, 1999

[4] J. Y. Lee, H. Kim, and S. B. Han, Web-Enabled Feature-Based Modeling in a Distributed Design Environment, Proceedings of the 1999 ASME-DETC Design for Manufacturing Conference, Las Vegas, Nevada, September 12-15, 1999

[5] R. Bidarra, W. F. Bronsvoort, Semantic Feature Modeling, Computer-Aided Design, Vol.32, 2000, pp. 201-225

[6] R. Gadh, R. Sonthi, Geometric Shape Abstraction for Internet-based Virtual Prototyping, Computer-Aided Design, Vol.30, No.6, 1998, pp.473-486

[7] Di Wu, Swati Bhargava, Radha Sarma, Solid Model Streaming as the Basis for a Distributed Design Environment, Proceedings of the 2000 ASME-DETC Design Automation Conference, Baltimore, Maryland, September 10-14th, 2000

[8] Jie Li, Shuming Gao, Xun Zhou, Direct Incremental Transmission of Boundary Representation, Proceedings of Eighth ACM Symposium on Solid Modeling and Application, June 16-20, Seattle, Washington, 2003

[9] Seungbum Koo, Kunwoo, Lee Wrap-around operation to make multi-resolution model of part and assembly Computers and Graphics (Pergamon); October 2002; v.26, no.5, p.687-700 [1] Adams, S.-H and Yang, M.-Y., A study on a generalized parametric interpolator with real-time jerk-limited acceleration, Computer-Aided Design, Vol. 36, No. 1, 2004, pp 27-36.