Incremental Error Analysis of 3D Polygonal Model through MAYA API

Prof. Yogesh Singh, Prof. B.V.R.Reddy, R.Rama Kishore
GGSIP University, Delhi

ABSTRACT
Generally applications in computer graphics use very high detailed models. These models are too compound for the limited hardware capacity and take much time to render and to transmit. Related fields can benefit from simplification of complex polygonal models. This introduces errors in the models during the process of simplification. It is require to judge when to stop the simplification process as rate of error change in the model is not same in every step of simplification process. It is required to measure the error in the model during simplification to judge the quality of the 3D model at every stage. It is proposed to measure the error in the model at every stage and analyze the rate of change of error in the model as a valuable tool for managing data complexity. This algorithm is implemented on 4 different sets of models. Each set contains models at different number of polygon levels. Experiments are repeated to measure error on them at each level. In order to gain in both memory and speed, VC++ API is developed and created a MLL (Maya link library) to load as a plug-in in Maya.

Key Words: Error metric, MAYAAPI, Plug–in.

I. Introduction
Computer graphics applications require very complex 3D models to preserve a convincing level of realism. More number of polygons is required to get competent realism in the model. But large amount of polygons causes difficulty in the applications where attributes like hardware capacity, processing time, transmission speed and rendering time are of most important concern [18]. Hence there is a need to simplify the 3D model. Simplification of a 3 D model means to reduce the number of polygons making that model. Simplification begins with a geometric description of an object and produces a new description that is similar in appearance to the original with few geometric primitives. But if we reduce the polygons the quality of model gets affected as there is a trade-off between quality and those 4 factors (processing time, hardware capacity, transmission time, rendering time). In 3D models many of the errors are introduced during the process of polygonal simplification. The error change is not uniform in the every step of total simplification process. Sometimes we need to apply some method to judge the quality of the new output with respect to the previous one [5].

Hence the focus of this work is to measure the error increment in the model at every step of the simplification process and analyze the error to help when to stop the simplification process. Thus optimization of simplification can be achieved. Because of the dependency of human vision on intuition and limitations of human persistence of vision a mathematical model is always preferred to be used as an error metric Human metric system is not very reliable and fails after certain limit. Thus a mathematical method is required to evaluate the error between models with the same accuracy and efficiency all the time [3, 19]. A consistent and quantitative method is required to get a consistent and accurate result again and again. Use of a good measuring error will always improve the final quality of the simplified model.

II. Maya API programming
The Maya API is a VC++ API that provides internal access to Maya and is available on the Platforms like Windows, Linux, Mac OS X and Irix[21,22],. One can use the API to implement two types of code resources that are plug-ins which extends the functionality of Maya and stand-alones such as console applications which can access and manipulate a Maya model. Plug-ins are built as dynamic or re-locatable libraries which are loaded into Maya using standard operating system functionality. Few important classes are used in this program are MFnMesh, MitDag, MDagPath etc[16].

III. Proposed error measurement method using MAYA Plug -in
It is proposed to analyze the amount of error introduction in the model at every step of simplification. Error is proposed to measure after every step in the simplification. The meaning of measuring error is to compare the given two models weather both look similar each other or not. In order to assess the quality of the model, some means of quantifying the notion of similarity is required in terms of geometry and as well as appearance. This paper uses geometry error to compare the models. In this work, a method which measures the average squared distance between the model with complex data ie original and simplified data ie simplified is used. So the error that is Ei will be defined as [17]:

$$E_i = \frac{1}{|X_n| + |X_i|}\left(\sum_{V \in X_n} d^2(V, M_i) + \sum_{V \in X_i} d^2(V, Mn)\right)$$

Where X_i, X_n are sets of points sampled on the models Mn (original model) and Mi (simplified) respectively. The distance d(v, M) is the minimum distance from v to the closest face of M. Using the above equation, the error will be calculated by comparing the quality the original, simplified models.

Algorithms being used

1. Extracting all polygonal model data like vertices, edges and faces from the original model(M_1) and store in a file1
2. Extracting all polygonal model data like vertices, edges and faces from the simplified model(M2) and store in a file2
3. Distance from point on M_1 to the all faces of M_2 is computed.
4. Minimum of these distances is squared and added to sum1
5. Step3 and 4 are repeated on all the points of model M_1
6. Steps 3,4 and 5 are done by changing M_1 to M_2 and M_2 to M_1 to calculate the sum2
7. Average of sum1 and sum2 with respect to total number of points on M_1 and M_2 is computed for error E_i.
8. Replace the model M_2 to M_1 and next simplified model as M_2 and repeat the procedure till the last stage of simplified model is considered.

IV. Experimentation

In this work, VC++ API is developed and created a MLL (Maya link library) to load as a plug-in through Maya plug-in manager. Plug-in is called through a MEL command. The error of the model is displayed in the out put screen in terms of scaled unit distance of MAYA.

Four different sets of models have been considered for in this work.

These models sets are:

1. Rubber Duck
2. Space Ship
3. Face
4. Bottle

4.1 Rubber Duck Model:

Duck model was simplified using proposed algorithm to 9 different levels and incremental error was measured through plug in and the details are as follows.

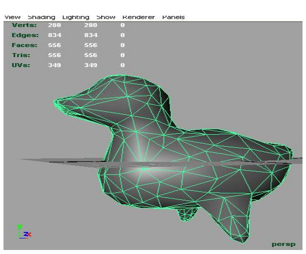

Fig 4..1.1 Original Duck model with 556 polygons

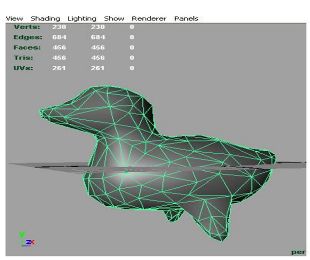

Fig 4.1.2 Duck model with 456 Polygons

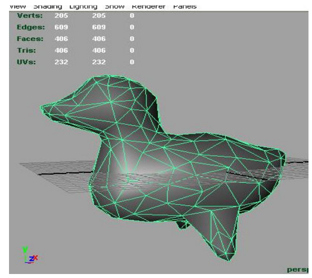

Fig 4.1.3 Duck model with 406 polygons

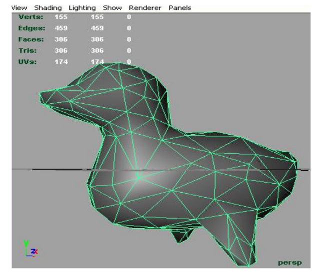

Fig 4.1.4 Duck model with 306 Polygons

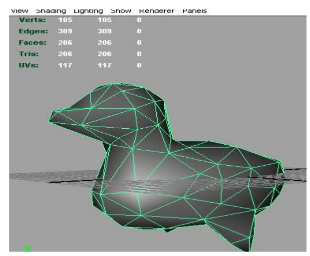

Fig 4.1.5 Duck model with 206 polygons

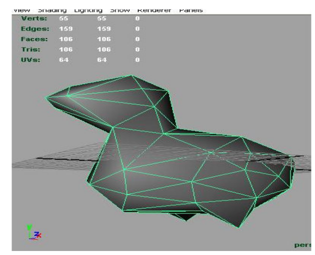

Fig 4.1.6 Duck model with 106 Polygons

Fig 4.1.1 to 4.1.6 shows Duck model simplified at different levels usin simplification

No of faces	change in Error E_i
556-456	1.9686E-05
456-406	0.000043
406-356	0.0000541
356-306	0.0001464
306-256	0.00027935
256-206	0.00052604
206-156	0.00207895
156-106	0.00166833
106-56	0.00939373

Table 4.1 No of faces versus Error change in Duck model

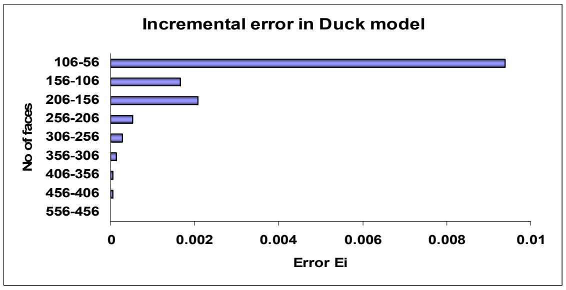

Fig 4.1.7 Rate of change in Error (Duck model)

4.2 Space Ship

Space ship model was simplified using proposed algorithm to 10 different levels and incremental error was measured through plug in and the details are as follows.

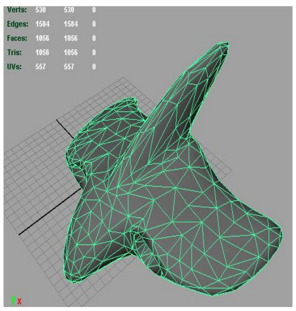

Fig 4..2.1 Original Space ship model with 1056 polygons

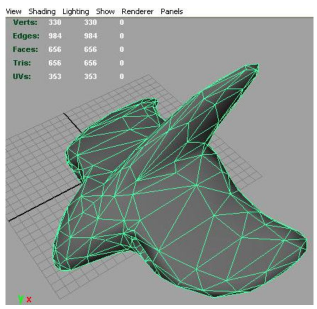

Fig 4.2.2 space ship model with 656Polygons

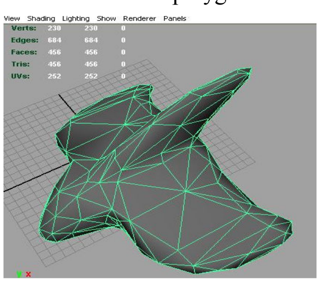

Fig 4.2.3 space ship model with 456 polygons

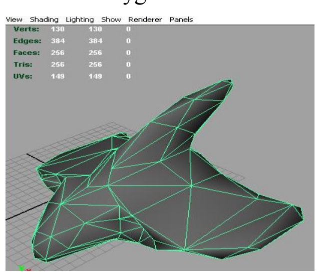

Fig 4.2.4 space ship model with 256 polygons

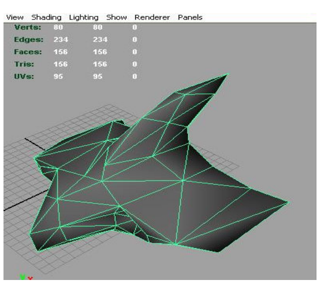

Fig 4.2.5 space ship model with 156 polygons

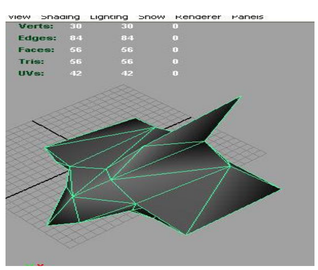

Fig 4.2.6 space ship model with 56 Polygons

Fig 4.2.1 to 4.2.6 shows space ship model simplified at different levels using simplification

No of faces	change in Error E_i
1056-956	0.000270105
956-856	0.000572944
856-756	0.000793604
756-656	0.00132817
656-556	0.00207818
556-456	0.00340192
456-356	0.00756641
356-256	0.015966
256-156	0.0729566

Table 4.2 No of faces versus Error change in Space ship model

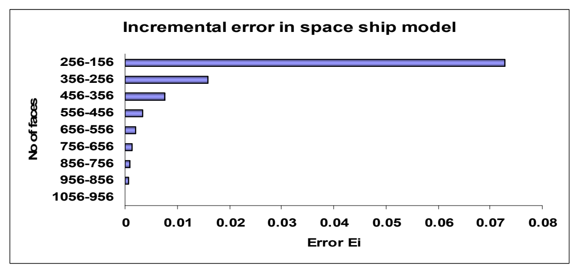

Fig 4.2.7 Rate of change in Error (space ship model)

4.3 Face Model:

Face model was simplified using proposed algorithm to 12 different levels and incremental error was measured through plug in and the details are as follows

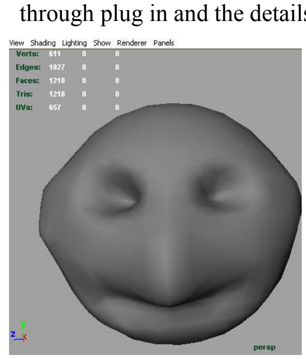

Fig 4..3.1 Face model with 1218 polygons

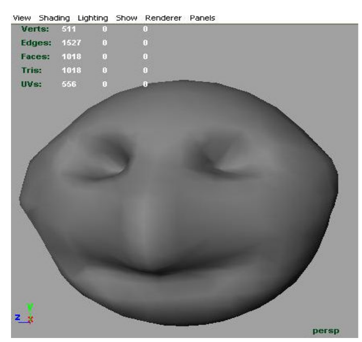

Fig 4.3.2 Face model with 1018Polygons

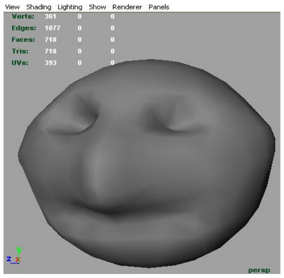

Fig 4.3.3 Face model with 718 polygons

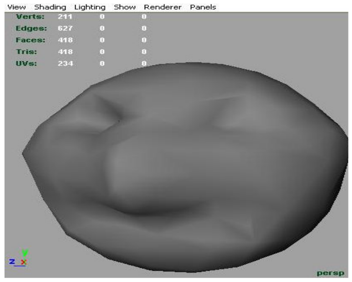

Fig 4.3.4 Face model with 418 polygons

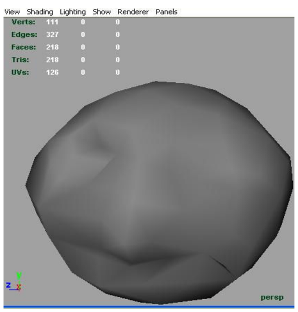

Fig 4.3.5 Face model with 218 polygons

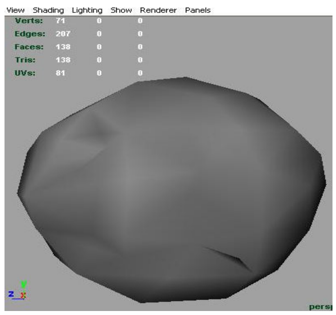

Fig 4.3.6 Face model with 138 Polygons

Fig 4.3.1 to 4.3.6 shows Face model simplified at different levels using simplification

No of faces	change in Error E_i
1318-1218	3.37189E-05
1218-1118	0.000066779
1118-1018	0.000111037
1018-918	0.000164917
918-818	0.000317539
818-718	0.000483444
718-618	0.000982902
618-518	0.00178176
518-418	0.00395626
418-318	0.00830856
318-218	0.0204892
218-138	0.0544295

Table 4.3 No of faces versus Error change in of Face model

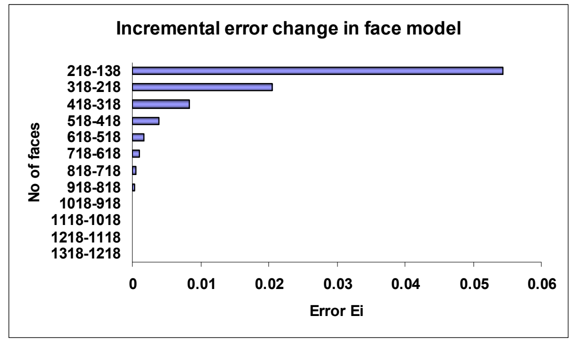

Fig 4.3.7 Rate of change in Error (Face model)

Bottle model:

Bottle model was simplified using proposed algorithm to 12 different levels and incremental error was measured through plug in and the details are as follows

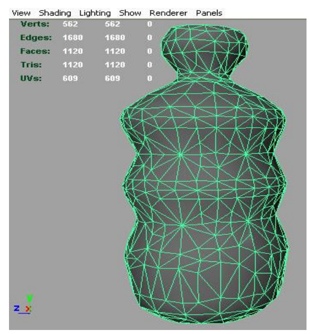

Fig 4..4.1 Bottle model with 1120 polygons

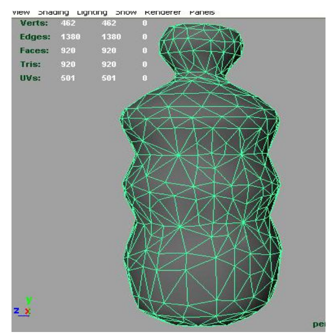

Fig 4.4.2 Bottle model with 920 polygons

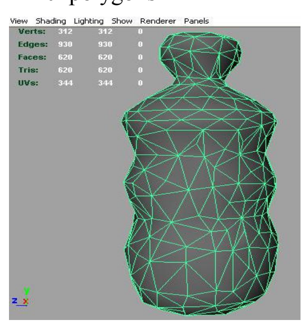

Fig 4.4.3 Bottle model with 620 polygons

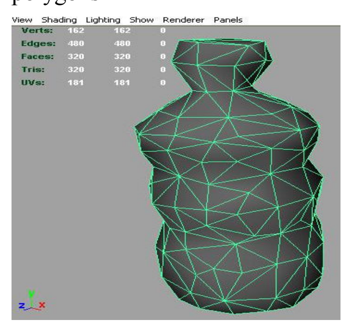

Fig 4.4.4 Bottle model with 320 polygons

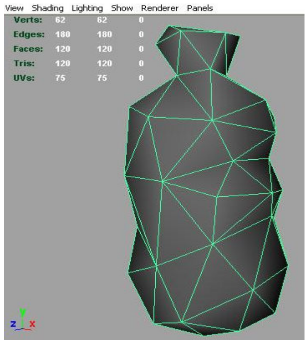

Fig 4.4.5 Bottle model with 120 polygons

Fig 4.4.1 to 4.4.5 shows Bottle model simplified at different levels using simplification

No of faces	change in Error E_i
1220-1120	0.000211787
1120-1020	0.000636225
1020-920	0.000820039
920-820	0.00126472
820-720	0.00180388
720-620	0.00273642
620-520	0.00485592
520-420	0.00823941
420-320	0.0182146
320-220	0.037664
220-120	0.16237

Table 4.4 No of faces versus Error change in Bottle model

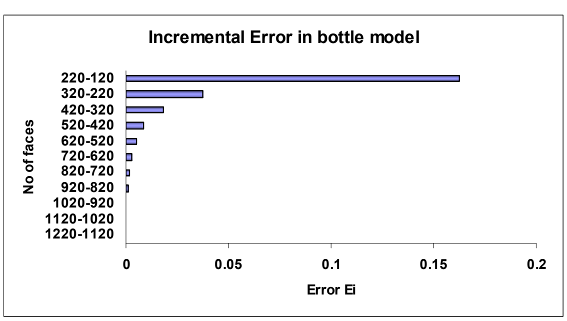

Fig 4.4.6 Rate of change in Error (Bottle model)

V. Conclusions

An incremental error in the models is analyzed. This is implemented through API programming and effectively used as a plug-in in MAYA. This can track the amount of error increase in the model while it is being simplified. This will help in deciding the optimized level of simplification during the whole process. It has been observed in all the models that as number of polygons are less the amount of error incurring is more even it is simplified slightly. The Error measurement is done at different levels with respect to recent past simplified model in the total simplification of model in each set. Four different sets are taken for the same. The results are convincing as the trend is found to be like error is inversely proportional to the no of polygons. Error is increasing as the numbers of polygons are decreasing. This approach of measuring error of 3D model is useful to determine the optimum level quality of any surface with greater speed and flexibility.

This method can be made utilized to optimize the working time on 3D model in respect to rendering time, transmission of 3D models over networks and 3D data compression. With this analysis models can be set at complexity of polygonal data that is reasonable to the user as per availability of the hardware capabilities.

VI. Future Direction

Improved Error Analysis: The incremental error Ei that has been taken as to evaluate the simplified model in terms of surface geometry. Similar approaches can be incorporated to evaluate the incremental error Ei considering similarity of appearance, color and texture values etc.

References

[1] Andr´e Gu´eziec. Surface simplification with variable tolerance. In *Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery (MRCAS '95)*, pages 132–139, November 1995.

[2] Alexis Gourdon. Simplification of irregular surface meshes in 3D medical images. In *Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed '95)*, pages 413–419, Apr. 1995

[3] Amitabh Varshney. *Hierarchical Geometric Approximations*. PhD thesis, Dept. of CS, U. of North Carolina, Chapel Hill, 1994. TR-050.

[4] A.M.Day,D.B. Arnold, S.Havemann, D.W. Fellner, "Combining Polygonal and subdivision Surface approaches to modeling and rendering of urban environments", *Computers &Graphics 28(2004) 497-507. ELSEVIER*

[5] Alan D. Kalvin and Russell H. Taylor. Superfaces: polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Appl.*, 16(3), May 1996

[6] Herv´e Delingette. Simplex meshes: a general representation for 3D shape reconstruction. Technical report, INRIA, Sophia Antipolis, France, Mar. 1994.

[7] Hugues Hoppe. Progressive meshes. In *SIGGRAPH '96 Proc.*, pages 99–108, Aug. 1996.

[8] Jihad El-Sana and Amitabh Varshney. Feneralized View-Dependent Simplification ,Volume 18,1999, EUROGRAPHICS

[9] Klein, Reinhard, Gunther Liebich, and Wolfgang Straßer, "Mesh Reduction with Error Control", *Proceedings of IEEE Visualization '96*

[10] L.H. You,Javier Romero Rodriguez,Jian J.Ahang.,"Manupulation of Elastically Deformable Surfaces through Maya Plug-in", *Proceedings of the Geometric Modelling and imaging- New Trends,IEEE2006*

[11] Michael Lounsbery. *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*. PhD thesis, Dept. of Computer Science and Engineering, U. of Washington, 1994.

[12] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95 Proc.*, pages 173–182. ACM, Aug. 1995.

[13] Marc Soucy and Denis Laurendeau. Multiresolution surface modeling based on hierarchical triangulation. *Computer Vision and Image Understanding*, 63(1):1–14,1996.

[14] Martin Franc, Vaclav Skala. Parallel Triangular Mesh Decimation With out Sorting. In 2001

[15] Muhammad Usman Keerio, Abdul Fattah Chandio, Attaullah Khawaja and Ali Raza Jafri , " Virtual Scene for Telerobotic Operation", *International Conference on emerging Trends IEEE2006*

[16] Maya 2008 manuals.

[17] Michael Garland and Paul Heckbert, "Surface Simplification using Quadric Error Metrics", *Proceedings of SIGGRAPH 97.* pp. 209-216.

[18] Paul Hinker and Charles Hansen. Geometric optimization. In *Proc. Visualization '93*, pages 189–195, San Jose, CA, October 1993.

.

[19] Reinhard Klein, Gunther Liebich, and W. Straßer. Mesh reduction with error control. In *Proceedings of Visualization '96*, pages 311–318, October 1996.

[20] R´emi Ronfard and Jarek Rossignac. Full-range approximation of triangulated polyhedra. *Computer Graphics Forum*, 15(3), Aug. 1996. Proc. Eurographics '96.

[21] Yun-Sang Kim, Sebastien Valette, Ho-You Jug, and Remy Prost. Local Wavelets Decomposition for 3-D Surfaces. In 1999

[22] Yacine Amara, Mario Gutiérrez, Frédéric Vexo and Daniel Thalmann, "A MAYA Exporting Plug-in for MPEG-4 FBA Human Characters", infoscience.epfl.ch/record/100258/files/Amara_and_al_Richmedia_03.pdf