ICID

ISSN 2088-5520

Proceeding

International Conference on Informatics for Development 2011

The Role Informatics for Achieving MDGs in Developing Countries

State Islamic University
Sunan Kalijaga
Yogyakarta, Indonesia
November 26th, 2011

Supported by

Information and Communication Technology
ict4d
INA FRONTIER
Indonesian Learning & Social Enterprise
UNESCO
United Nations Educational, Scientific and Cultural Organization
uniTwin
UNESCO Chair in ICT for Development
ict for All Club
APTIKOM
LA TROBE UNIVERSITY
AUSTRALIA

Preface

Assalamu'alaikum Wr. Wb.

The International Conference on Informatics for Development (ICID) aims to bring together academicians, researchers and practitioners interested in the application of Information and Communication Technologies (ICT) to many aspects of life in general and to the specialized support to person's daily life in particular. Government, education, health, business and industry are just few sectors currently used ICT to improve their business processes.

Standardization of procedures, strategies for adoption, and selection of alternative technologies are some of the issues that informatics and the ICT in general need to address in order to further promote ICT for development. In the case of developing countries, the use of ICT has had important results in the enhancement of the quality of life, contributing to a full integration of all citizens in the societies to improve their quality of life.

As one of the emerging research fields, ICT should be able to address crucial problems currently faced by many unfortunate people throughout the world. Hence, the conference will bring up a theme on "The roles of informatics for achieving Millenium Development Goals (MDGs) in developing countries."

The interest for this conference is quite high. It is reflected by the number of authors who have been submitting their papers. They are coming from different countries located in various continents: North America, Europe, Asia, Africa, and Australia. It all has not been possible without the contributions from our keynote speakers, authors, reviewers, and participants. I would also like to take this opportunity to express my appreciation to the steering and organizing committee. Finally, I wish you to enjoy this conference and our beautiful city: Yogyakarta.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, November 26th, 2011

Prof. Drs. H. Akh. Minhaji, M.A., Ph.D
Dean, Faculty of Science and Technology, State Islamic University Sunan Kalijaga

CONFERENCE COMMITTEE

Keynote Speaker

1. Dr. Dorothea Klein; ICT4D Centre/UNESCO Chair in ICT for Development; Lecturer, Royal Holloway, University of London
2. Assoc. Prof. Wenny Rahayu; Lecturer, La Trobe University
3. Prof. (Dr.) Srikanta Patnaik; President, Interscience Research Network (IRNeT), India

Scientific Committee

1. Prof. Richard Heeks; The University of Manchester
2. Prof. Deokjai Choi; Chonnam National University
3. Prof. Jongtae Rhee; Dongguk University
4. Assoc. Prof. Abdullah Saad Almudimigh; King Saud University
5. Assoc. Prof. Zarina Shukur; Universiti Kebangsaan Malaysia
6. Dr. Hj. Mohammad Nabil Almunawar; University of Brunei Darussalam
7. Assoc. Prof. Retantyo Wardoyo; Universitas Gadjah Mada
8. Prof. Kuspriyanto; Institut Teknologi Bandung
9. Dr. Agus Zainal Arifin; Institut Teknologi Sepuluh Nopember

Steering Committee

1. Prof. Dr. H. Musa Asy'arie (Rector of UIN Sunan Kalijaga Yogyakarta)
2. Prof. Drs. H. Akh. Minhaji, M.A., Ph.D (Dean, Faculty of Science and Technology, UIN Sunan Kalijaga Yogyakarta)
3. Khurul Wardati, M.Si (Deputy Dean on Academic Affairs, Faculty of Science and Technology, UIN Sunan Kalijaga Yogyakarta)
4. Agus Mulyanto, M.Kom (Head of Informatics Department, UIN Sunan Kalijaga Yogyakarta)

Organizing Committee

Chair : Agung Fatwanto, **Co-Chair** : Bambang Sugiantoro, **Secretary** : Sumarsono & Shofwatul 'Uyun, **Treasury** : Maria Ulfah Siregar & Latifah, **Venue Chairperson** : Ade Ratnasari, **Publicity Chairperson** : Nurochman, **Proceeding Chairperson** : M. Didik R. Wahyudi, **Program Chairperson** : M. Mustaqim, **Logistic Chairperson** : Aulia Faqih Rifai, **Session Chair** : M. Anshari & M. Taufiq Nuruzzaman, **Website Chairperson** : Muhammad Rifqi M, M. Alex Syaekhoni, & Imam Fahrurrozi, **Technical Supports** : Yusuf Murdani, M.Iqbal, M.Munawir, Awan Pramudya, & Iriyanta

Alamat Redaksi dan Penerbit : Program Studi Teknik Informatika, Fak. Sains & Teknologi UIN Sunan Kalijaga Jl. Laksda Adisucipto, No.1 Yogyakarta Telp (0274) 519739, Faks.(0274)540971. Website : http://www.icid2011.com/ Email : committee@icid2011.com

CONTENTS

Cluster 2 : Artificial Intelligence and Image Processing

Cluster – 3 : Computer System and Network

Cluster 4 : Database and Information System

The Use of LOD (Level of Detail) For Optimizing A Web3D Based Site

Case study: Web3D Site of UAD's 3rd Campus

Mursid W. Hananto
Information Systems Department, Faculty of Mathematics & Natural Science
Ahmad Dahlan University
Yogyakarta
toxynusca@yahoo.com

Abstract—**On a website which used Web3D visualization style just like described by Web3D Consortium on their standards, the more polygons used and more complexity on objects incorporated inside it will make the website goes slower and heavier. Visitors of such website will have to wait for a considered long time before they can have a full display of the visited website, and it might have to wait for another more time before the website can actually accept any responses from the visitors. This happened also on UAD's 3rd Campus Web3D prototype site which was used as a test-site to look for the most suitable optimizing methods currently available. LOD method partially used on the site has shown some good effects on the website, but it also shown some unwanted visuals and other aspects which might degrade the overall visual quality of the website. This paper presented the results and data collected from the attempts to use LOD method on the UAD's 3rd Campus Web3D site which made heavy use of complex polygons to create more realistic representation on the Internet. The facts collected suggested that LOD might be used as a good optimization method with some exception on some specific aspects to keep the site navigation experience high (in user's perspective).**

Keywords: LOD; optimization; complex; polygon; Web3D

I. Introduction

Web3D sites area commonly made from polygon objects. Other primary elements including raster images which mainly act as the surface of the polygons to create the skins that will make the objects look better than just a plain surface. Elements like independent points and lines are not often used in a Web3D scene, since a Web3D world generally tried to represent physical objects from the real world which rarely simple enough just to draw them by the use of points and lines.

To make the Web3D site creation much simpler but resulting in faster and responsive site, Web3D scenes could make heavy use of raster images instead of complex and detailed polygons. The Temple of Ikaros developed by Abdul Muttaleb Balam [1] is an excellent Web3D site that used many raster images for this purpose. Advantages of this method including shorter development time, simpler objects needed, easier polygon creation, and most of all, a high speed navigation inside the created world. Visitors can enjoy an accelerated navigation since the browsing computer only need to render simple objects and paint the raster image as the skin to the objects surfaces. The only added workload will come from additional effects used, such as lighting, shading, animated texture, complex texturing, animated objects, and some other 3D effects available for Web3D worlds. Fortunately, these additional workloads have been handled by the current advancement of 3D graphics accelerator technology which has the capability to tackle almost all kind of complex 3D graphics rendering and effects currently known in this area.

All the good aspects of this method are opposed by the drawback in visual which recognized almost immediately by visitors after they made a brief navigation inside the site. Lack of details on the objects presented on-screen will make visitors felt like they were watching something that is far from realistic. Even if the surfaces seemed good on the first glimpse, the next sight will made them realize that the shapes of objects were fake ones. When visitors view the objects from afar, it might looked normal and favorable, but as they are getting closer to the objects presented, they will see that most things are in a simple flat shaped forms. Textures used were sometimes low in resolution, and this will only worsen visitors' navigation experience gathered from the visited Web3D site.

Other way to develop a Web3D site is to utilize as many polygon as possible, to create a closer to realistic object representation when viewed by visitors. This method was used on the development of UAD's 3rd Campus Web3D site as described in [2]. By applying actual shape detail in the form of more complex 3D objects rendered inside the Web3D scenery (also called "world", a similar term to "page" when compared to conventional 2D web document), overall Web3D site will have a constantly good visual whether the objects were seen from afar or from a close view. Raster images can still be used but they will serve as an actual skin rather than to show the virtual shape of the objects. Some flat and repeated surface pattern (such as floor tiles) can also make a good use of raster images to make the scenery much more realistic and closer to its represented objects in the real world.

Using so many detailed polygons in a Web3D scenery usually will reveal some unwanted side-effects which might clearly presented from the first seconds when visitor's browser start to display website's contents on the screen. The first noticeable effect is that more time needed to load the scenery. Even when visitors are using a high-speed Internet connection along with a good set of computer hardware (compliant to the current level of processor, memory, and 3D graphics), they will immediately notice that the target website seemed so slow and took a longer waiting time to be viewed by visitors (compared to conventional 2D site or Web3D with simpler scenery).

The second common effect will rise after the site has been loaded. Navigation around objects inside the world will look sluggish, and sometimes visitors will feel that they are browsing with some heavy loads on their computers. These experiences are common on a world (or distributed world) which has many complex objects with all the details on them. Some sites even displayed incorrectly inside the visitors' browsers, not only because there were errors on the document codes but they occured because of browser misinterpretation to the codes as the results of incomplete transfer of entire documents required to create the scenery.

The same effects have also been found on the UAD's 3rd Campus experimental Web3D site which applying polygons on most of its viewable objects. Unlike most Web3D sites which still prefer to use raster images to visualize many small details that considered as "might only be seen at a glance", every objects in the UAD's 3rd Campus site were complex and detailed polygons which made from scratch and merged together in a locally distributed world documents. A sample screenshot of complex objects used in the site is depicted in Fig. 1 below which represents detailed window bars (trellis) complete with its surface effect to add a metallic appearance.

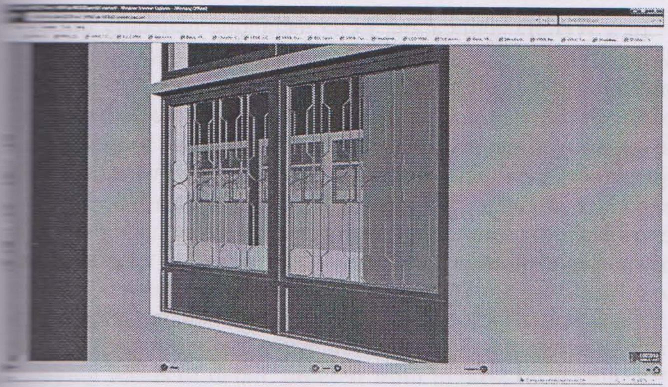

Figure 1. A sample window bars object used on scenery

Although the website has undergone an exhaustive code optimization which reduced its overall size from about 250 Megabytes to only about 23 Megabytes (including raster images which served as surface textures) and then compressed as wrz files (as opposed to uncompressed wrl files) which reduced the size to a smaller 15 Megabytes, the website still appeared slow and heavy both before and after the loading process. On a computer equipped with dual-core processor which has 2 GigaHertz FSB clockspeed frequency, 1 Gigabyte memory, and 256 Megabytes video framebuffer on a 3D accelerator capable graphics adapter, it runs quite slow on its default setting, with a range of dynamic rendering performance measured in 10-20 frames per second. Depending on navigation mode selected and visible objects shown on the visitor's browser screen, these values can drop down to below 10 frames per second which was deemed unsuitable for a smooth 3D navigation.

The UAD's 3rd Campus Web3D site would clearly needed some tweaks to make it more suitable for any walkthrough navigation on any location inside it and still keeps the details already made into the scenery. Depicted in Fig. 2 below is the first actual display after the loading process completed successfully, based on preset default world camera view.

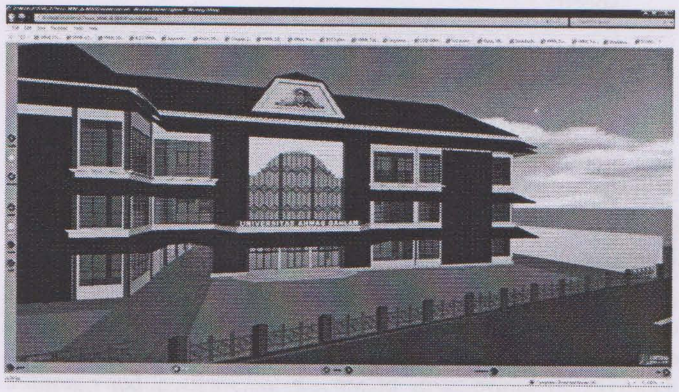

Figure 2. Opening view of the UAD's 3rd Campus world

II. Level Of Detail

Schmalstieg, and Schaufler in [3] stated that when rendering complex three-dimensional scenes, it is commonly the case that many objects are very small or distant. The size of many geometric features of these objects falls below the perception threshold or is smaller than a pixel on the screen. To better use the effort put into rendering such features, an object should be represented at multiple levels of detail (LODs). Simpler representation of an object can be used to improve the frame rates and memory utilization during interactive rendering. Levels of detail (LODs) are used in interactive computer graphics to avoid overload of the rendering hardware with medium to high numbers of polygons.

A. LOD Techniques

Currently, there are so many LOD techniques existed. Research on this area are still conducted by many parties in order to achieve a better techniques for a common or specific case. One of the LOD approaches is proposed by Agrawal et al. [4]. This approach is to reduce the scene complexity without leading to an inferior visual representation. The method derived was using multiresolution models to provide different LOD representation of the modeled objects. They have have developed a view-dependent dynamic block-based LOD mesh simplification scheme and out-of-core management of large terrain data for real-time rendering on desktop PCs. The method can be used to demonstrate a new geometry-based mapping approach to render map vector data over multiresolution 3D terrain model.

There are 3 basic frameworks for managing LOD according to Luebke et al. [5]: discrete, continuous, and view-dependent LOD.

1) Discrete LOD: a scheme used in most 3D graphics applications today, this approach creates multiple versions of every object, each at a different level of detail, during an offline preprocess. At run-time the appropriate LOD is chosen to represent the object. Advantages of this method includes:

- Total number of polygons is reduced since the distant objects use coarser LODs, thus it speeds up rendering speed.
- Decoupling simplification and rendering makes this the simplest model to program.

2) Continuous LOD: rather than creating individual LODs during preprocessing stage, the simplification system create a data structure encoding a continuous spectrum of detail. The desired LOD is then extracted from this structure at run-time. Advantages of this method includes:

- Better granularity, since the LOD for each object is specified exactly rather than selected from a few precreated options.
- It will only use as many polygons as needed for the desired LOD, and frees up more polygons for other objects.
- Continuous LOD also supports streaming of polygonal models, enabling progressive rendering properties.

3) View-Dependent LOD: this extends Continuous LOD, using view-dependent simplification criteria to dynamically select the most appropriate LOD for the current view. View-dependent LOD is anisotropic, a single object can span multiple levels of simplification. Advantages of this method includes:

- A better granularity : polygons are allocated where they are most needed within objects, as well as among objects.
- Better fidelity for a given polygon count, optimizing the distribution of this scarce resource.
- It can enable interactive rendering without manual intervention or extra processing for segmentation.

LOD can also be used to ease a wide surface rendering workload. As stated by Hugues Hoppe in his paper [6], real-time visualization of large-scale surfaces is a challenging problem. The most common approach for rendering such surfaces is to exploit the traditional 3D graphics pipeline, which is optimized to transform and texture-map triangles. In fact, the surface usually exhibits significant spatial coherence, so that its perspective projection can be approximated to an accuracy of a few pixels by a much simpler mesh. Finding such a mesh, and updating it as the viewing parameters change, is referred to as view-dependent level-of-detail (LOD) control.

Several schemes have been developed to address view-dependent LOD control. Among those, the view dependent progressive mesh (VDPM) framework [7] represents an arbitrary triangle mesh as a hierarchy of geometrically optimized refinement transformations. Hoppe proposed to extend VDPM framework into two areas:

a) Memory requirements: Redesigning the data structures to be output-sensitive, thereby reducing memory requirements.

b) Runtime geomorphs: Introducing a scheme for efficient runtime creation of geomorphs, which smoothly transition surface geometry over several frames to eliminate popping.

B. LOD in Web3D Applications

As described by Pasman and Jansen in [8], appropriate levels of detail (LOD) can maximise the visual quality of the rendering while minimising the rendering costs. Especially in thin client systems, simple and meshed imposters are very good in reducing the number of polygons and the resource usage in the client, while preserving the quality of the final images.

Pasman and Jansen in [8] also stated that simplified polygon models are a good choice to replace nearby objects because the observer can move around them freely without requiring the system to refresh the model. But for distant objects this is overly accurate, as the observer will probably never see the back faces. If the polygon model is simplified extremely, the simplification will become very distorted. Instead, simple imposters holding a picture of the object are much better when it comes to simplifying objects to a very low number of polygons. Meshed imposters are most useful at moderate distances.

In a Web3D application, LOD technique generally utilized by specifying some definition layers, each layers contains the limit/threshold required to automaticaly trigger the specific codes or objects which then displayed on the screen, replacing the other layer that previously placed on the same location. The replacement view will considerably decrease the workload that has to be computed, compared to the original detailed view. As described in [9], the application developed was using 4 LOD approaches to reduce the amount of information displayed on the screen:

a) Layer selection: Manual selection of the required 2D layers for each LOD.

b) Mesh simplification algorithm: Edge contraction algorithm.

c) Edge factor: large penalty factor in order to preserve the boundaries of the 2D layers.

d) Layer generalisation: Using Douglas-Peucker algorithm.

The LODs used were tiles with different measurement, from 500m to 64000m, and the layers used were varied from 105 to 6 layers. The closer LOD would have the bigger byte size and the smaller error simplification.

III. Implementing the LOD

Overall, the LOD technique selected for this research was the Layer Selection as described in [9]. Manual selection was chosen since there were many objects in the scene which used uniform pattern for the entire view inside the world document but needed to act differently based on viewer's distance. This selection could be embedded on each separated document which contain the definitions for each object used on the scenery.

A. LOD in VRML

UAD's 3rd Campus Web3D scenery was developed using Virtual Reality Modelling Language (VRML) format. According to Mark Pesce in [10], VRML is a technology which is able to bring 3D shapes e.g. architecture, room, and space to the Web. VRML started from 1.0 version, followed by VRML 2.0 in 1996 with many additional features, and then replaced with VRML97 in December 1997 which added new features to complement the previous version. UAD's 3rd Campus Web3D is created using the VRML97 standard described in ISO/IEC 14772-1:1997 document according to VRML97 Specifications from VRML/Web3D Consortium).

Based on the Annotated VRML97 Reference Manual by Rikk Carey and Gavin Bell, LOD node is typically used for switching between different versions of geometry at specified distances from the viewer. However, if the range field is left at its default value, the browser selects the most appropriate child from the list given. It can make this selection based on performance or perceived importance of the object. Children should be listed with most detailed version first just as for the normal case. This "performance LOD" feature can be combined with the normal LOD function to give the browser a selection of children from which to choose at each distance.

```
LOD {
  exposedField MFNode  level    []
  field        SFVec3f center   0 0 0    # (-,)
  field        MFFloat range    []       # (0,)
}
```

The LOD node specifies various levels of detail or complexity for a given object, and provides hints allowing browsers to automatically choose the appropriate version of the object based on the distance from the user. The level field contains a list of nodes that represent the same object or objects at varying levels of detail, ordered from highest level of detail to the lowest level of detail. The range field specifies the ideal distances at which to switch between the levels.

The center field is a translation offset in the local coordinate system that specifies the centre of the LOD node for distance calculations. The number of nodes in the level field shall exceed the number of values in the range field by one (i.e., N+1 level values for N range values). The range field contains monotonic increasing values that shall be greater than 0. In order to calculate which level to display, first the distance is calculated from the viewer's location, transformed into the local coordinate system of the LOD node (including any scaling transformations), to the center point of the LOD node. The LOD node evaluates the step function L(d) to choose a level for a given value of d (where d is the distance from the viewer position to the centre of the LOD node).

A sample code to implement the LOD for this purpose is as follows:

```
#VRML V2.0 utf8
LOD {
    range [ 10, 50 ]
    level [
        LOD {
            level [
                Shape { geometry Sphere { } }
                DEF LoRes Shape { geometry Box { } }
            ]
        }
        USE LoRes,
        Shape { }
    ]
}
```

In this code, the browser is free to choose either a detailed or a less-detailed version of the object when the viewer is closer than 10 meters (as measured in the coordinate space of the LOD). The browser should display the less detailed version of the object if the viewer is between 10 and 50 meters and should display nothing at all if the viewer is farther than 50 meters. Browsers should try to honor the hints given by web authors, and authors should try to give browsers as much freedom as they can to choose levels of detail based on performance.

As in UAD's 3rd Campus Web3D, the LOD node can be used in the following way:

- if the user is close to the object draw the most detailed version
- when the user is not close anymore, but still not to far away draw a less detailed version
- when the user is very far away draw only a crude version of the object.

By selecting less detailed versions when the user is not close to a object, time is saved and the user perceives no difference due to distance. Levels of detail can be specified as many as desired . Authors should try to keep changes from one level of detail to the next as small as possible so that a performance break will not occur. The LOD can also be used to avoid drawing objects which are invisible, for instance in another room. In this case authors can specify an empty object, for instance a Shape without a geometry.

The range fields specify which version of the object are drawn. Objects should be specified in the level field by decreasing LOD. If the distance from the visitors to the object is smaller than the first range specified, then the first version, the more detailed one, will be drawn.

B. Utilization on the test-site

LODs were used in the 3rd Campus world in these categories of usage:

- For small repeatable uniformed objects which viewable from many points of view inside the 3rd

Campus world (e.g. window sets and doors), the LODs were used in 3 levels:

- viewed in detail when the visitors camera reached below 5 meters from the objects,
- replaced by less detailed (much simpler) objects when the camera in the range of 5 to 10 meters, and
- totally replaced with their related raster images embedded on flat surfaces when visitor's camera got farther (more than 10 meters).

- For small repeatable uniformed objects which were not viewable except the visitors' displayed them directly on their planted locations (e.g. students' chairs, tables, and other room objects), the LODs were used in 2 levels:
  - Viewed in detail when the visitors camera entering the room which contained the objects, (proximity range 10 meters).
  - Invisible by visitors' camera when they move out from the room, removed from the screen when visitors's camera got farther than 10 meters.
- For large independent objects (e.g. mosaic glass window and announcement board), if the object has complex and detailed shape then it will be replaced by a much simpler object without any raster surface. If the object is a simple 3D form (e.g. floor plates) then there's no replacement level if visitors' camera got farther (more than 20 meters).

Rebuilding of the UAD's 3rd Campus world were not just re-encoding the 3D objects which will adopt the LOD instrustions, but also creating replacement objects related with the original detailed objects. Other works including creating raster images to be embedded on flat surfaces. To speed up the creation, raster images were created from screen-captured images from the original detailed objects. Another additional work is to place the raster images to their locations so that they can exactly replace the view from the original detailed objects from afar. The work involving converting the objects document format so that the objects could be edited using a 3D modeler software and then re-exported back to VRML format to be edited by a text editor software.

IV. RESULTS AND DISCUSSIONS

After the addition of LOD codes completed, the site can be used to see the LOD effects added on some various views. The LODs has been placed according to the rules as stated above, so the results were varied according to the positions taken when the tests were carried out. The former copy of the world document (without any LODs) were tested first, and then the modified version took place with the same test scenarios.

The tests were taken on two type of browsing mode: directly browsing from the computer that also store the files which form the site, and remote browsing which carried out from a different computer which is connected to the host computer via wired Local Area Network (LAN). It is important to highlight that Web3D browsing mostly done by downloading all components to local (visitors') computer, that means after the sites' 3D components have been thoroughly downloaded to visitors' computer, there's almost no need to stay connected to the hosting computer.

A. *Effects on world navigation*

First loading of the world document was much shorter (about 4 seconds) than the former site when viewed remotely, compared to almost no differences when loaded on locally hosted computer. After that, both tests yielded almost identical result except when the viewing remote computer needed to get the data from the host computer.

There were notable increase of speed and smoothness in all modes of navigation used (Walk, Fly, and Study), and it was also shown on the frame per second (fps) measurement that has an addition in values even it looked not too high (varies about 5 to 20 fps). But in some places where the LOD proximity level is in effect, there were stoppages that took place even just in split second before the more detailed view replace the scenery. The stoppage did not happened if the camera walk away farther and the scenery was replaced by a less detailed view. At first, it was thought that the stoppage would only happened at the first loading of the viewed objects. When the same navigation event got repeated, the stoppage always took place. This has not been investigated yet, but there was a possibility that it might caused by the viewing browser which was unable to load the scenery fast enough.

Aside from any stoppages or delays when the browser camera moved to LOD-affected positions, overall navigation were fluent and there were increase in navigation speed (the default speed was not changed) compared to the original world. Overall navigation experience also got slight better than a world without LODs.

B. *Effects on display quality*

On the other hand, display quality was not increased, at least from the testers' comments, the ones who navigate the site and got the navigation experience first hand. Some testers even noticed a slight degradation on overall display quality when they tried to observe some objects from afar. Not all testers were so thorough when they navigate the scenery of UAD's 3rd Campus, but when they did, sometimes they saw unrealistic view such as entirely flat window or a window that exhibit the same view even when the testers' camera have already shifted to a different positions.

For a quick navigation, the slight decrease in appearances were not disturbing. But for a close and long observation, visitors' might stumble across low quality views which were supposed to be thrown away by the use of detailed objects. The close-to-realistic view might be degraded, but for a faster and smoother navigation experience, these decrease can be omitted if there's no absolute need for perfect and realistic view at all navigation times.

C. Display anomalies

Although it might caused by incorrect placement on the coding phase, some raster images did mistakenly placed. Some 3D objects also shown incomplete, even when the same objects viewed from other locations. The world documents originally were a distributed type Web3D site, which constructed from many smaller worlds as its components. These objects were called if there was any view that need their presence. It was unknown that the same object could appear different even it came from the same code.

Anomalies found were only occured on some specific places which have a complex combination with other objects. The same anomaly were not occured on the original world, but some small anomalies from the original world were not found on the modified world. These were noted as a new problem that needed to be investigated on a further research.

D. Performance Evaluation

Below are the recorded results from the tests commenced to evaluate the performance of LOD used on the site.

TABLE I. COMPARISON OF VALUES

MEASURED ELEMENTS	w/o LOD	w/ LOD
Average fps (walk)	13,3 fps	22,4 fps
Average fps (fly)	12,1 fps	25,7 fps
Average fps (study)	9 fps	16 fps
Average loading time (local)	9,1 s	8,7 s
Average loading time (remote)	14,2 s	10,5 s
Average delay on viewing	0,3 s	1,8 s

The results mentioned in Table I above only stated the technical values measured directly from both sites. There were increase in performance, except for the higher delay on the site with LOD. Table II below were the average opinion scores from 5 testers, gathered from both local and remote tests when they compare their own experiences both on the original Web3D site and on the modified Web3D site.

TABLE II. COMPARISON OF TESTERS' OPINIONS

STATEMENTS	Better	Worse
Overall navigation speed experience	10	0
Overall visual experience	8	2
Close range visual observation	9	1
Medium range visual observation	8	2
Long range visual observation	5	5
Loading time waiting experience	10	0
Visual delay experience	3	7
Display anomalies experience	4	6
Stoppages experience	2	8

Based from these results, overall speed experience is considered higher, including when the testers observed a room as depicted in Fig. 3 below, but some visual trade-offs went below the expectation. The utilization of LOD needed more adjustments to adapt the unique problems of tested Web3D site which utilizes complex polygons on all type of shapes elaborated inside the scenery except for the flat surfaces.

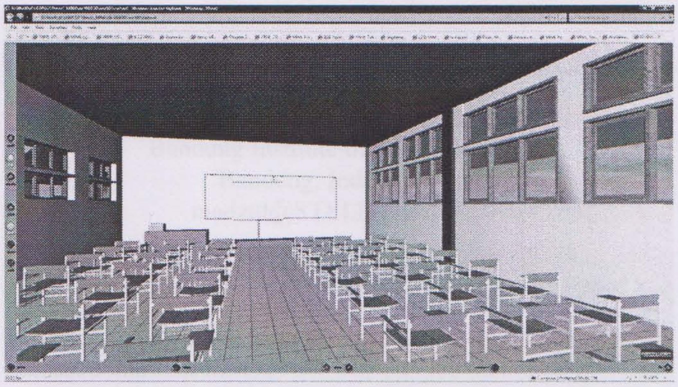

Figure 3. A sample classroom scenery

The use of view range limiters and optimized lighting should also be considered for the next experiment in order to further enhance the viewing quality and experience of the visitors when they navigate inside this Web3D site.

ACKNOWLEDGMENT

Author wishes to thank Agus Harjoko of the Department of Computer Science and Electronics of Gadjah Mada University for the guidance in the development phases of Web3D site used for this research, also to Imam Azhari and Suprihatin of the Information System Department of Ahmad Dahlan University for the support and allowance to utilize department's resources for this research.

REFERENCES

[1] A.M. Ballam, "Resurrecting Ikaros", Thesis, University of California at Los Angeles, College of Art and Architecture, Department of Architecture, Summer 1999.

[2] M. W. Hananto, "Visualization of UAD's 3rd Campus in Web3D Using VRML Format", Thesis, Gadjah Mada University, Faculty of Mathematics and Natural Science, Computer Science Post Graduate Program, 2007.

[3] D. Schmalstieg, and G. Schaufler, "Smooth Levels of Detail," in *Proceedings of VRAIS'97* (March 1–5 1997), IEEE, pp. 12–19, 1997.

[4] A. Agrawal, M. Radhakrishna, R.C. Joshi, "Geometry-based Mapping and Rendering of Vector Data over LOD Phototextured 3D Terrain Models," WSCG 2006 - The 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Jan. 30 – Feb. 3, 2006, Plzen - Bory, Czech Republic, 2006.

[5] D. Luebke, M. Reddy, J.D. Cohen, A. Varshney, B. Watson, and R. Huebner, Level of Detail for 3D Graphics. San Fransisco, CA: Morgan Kauffman Publishers, 2003.

[6] H. Hoppe, "Smooth view-dependent level-of-detail control and its application to terrain rendering," IEEE Visualization Conference, pp. 35-42, October 1998.

[7] H. Hoppe, "View-dependent Refinement of Progressive Meshes," Computer Graphics (SIGGRAPH '97 Proceedings), pp 189–198, 1997.

[8] W. Pasman and F.W. Jansen, "Scheduling Level of Detail with Guaranteed Quality and Cost," in Proceedings of 7th International Conference on 3D Web Technology, pp 43–51, Tempe, Arizona, February 2002.

[9] M. Over, A. Schilling, S. Neubauer, S. Lanig, and A. Zipf, "Processing Mass Data for a Nationwide Web3D Service," in Proceedings of 12th AGILE International Conference on Geographic Information Science, Leibniz Universität Hannover, Germany, 2009.

[10] M. Pesce, VRML: Browsing & Building Cyberspace. Indianapolis: New Riders Publishing, 1995.

Organized by:
Informatics Department
Faculty of Science and Technology
State Islamic University Sunan Kalijaga
Yogyakarta, Indonesia

9 772088 552009