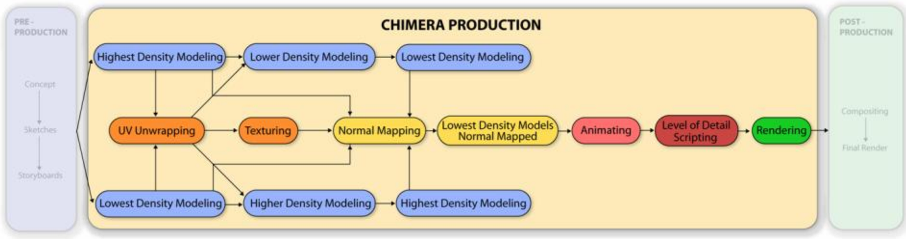
Chimera: A Hybrid Pipeline for Film and Broadcast Animation Production



Justin Blignaut
Dept. Computer Graphics Technology
Purdue University
jblignau@purdue.edu

Bryce Hyland
Dept. Computer Graphics Technology
Purdue University
bhyland@purdue.edu

Kurt Luther
Dept. Computer Graphics Technology
Purdue University
lutherk@purdue.edu

Kevin O'Keefe
Dept. Computer Graphics Technology
Purdue University
kokeefe@purdue.edu

December 14, 2005

Chimera: A Hybrid Pipeline for Film and Broadcast Animation Production

Justin Blignaut
Dept. Computer Graphics Technology
Purdue University
jblignau@purdue.edu

Bryce Hyland
Dept. Computer Graphics Technology
Purdue University
bhyland@purdue.edu

Kurt Luther
Dept. Computer Graphics Technology
Purdue University
lutherk@purdue.edu

Kevin O'Keefe
Dept. Computer Graphics Technology
Purdue University
kokeefe@purdue.edu

Abstract

Small production houses in the film and broadcast animation industry lack the resources to render complex 3D scenes which require massive data sets. Consequently, time and money is wasted on hodgepodge techniques that look less convincing. Chimera, a hybrid animation production pipeline, combines normal mapping and level of detail to drastically improve run-time rendering performance while maintaining perceived visual quality. As a result, small production houses can render animations that look more detailed using software and hardware they already own.

1. Introduction

There is a tremendous need for the development of a production process to manage and render extremely large data sets (Cohen, 2005). While research-oriented visualization science has produced techniques and procedures for manipulating and rendering huge quantities of data, the migration of this technology to practitioners' tool sets has been slow or nonexistent. For example, radiosity was invented in 1984 at Cornell University (Coral, Torrance, & Greenberg, 1984) and saw widespread use in the scientific visualization field by the late 1980s, but did not appear in commercial animation packages until a decade later. The same pattern can be repeatedly observed with computer graphics techniques such as subdivision surfaces, UV mapping, and so on. With respect to the development of novel rendering techniques, this lag time between research in academia and implementation in industry represents a discrepancy with potentially significant economic consequences.

Comment [BH1]: Do you have any citations?

An important example of this discrepancy exists between the data management and rendering techniques used for film and broadcast animation, and real-time interactive 3D techniques used to render massive data sets. When producing animations for film and broadcast, it is common practice to render animations in layers (such as background and foreground elements, specular and diffuse maps, etc.) and composite them to create the final frames; all of the elements seen on a final frame are not rendered simultaneously. One purpose of this technique is to reduce the computational expense of rendering the massive amounts of data required to produce a completed image. People working in the field of 3D interactive graphics have taken a very different approach. By using techniques such as normal mapping and level of detail, they have been able to manage and render large data sets in an efficient manner for real-time rendering (Shaffer & Garland 2005).

Comment [BH2]: Again, citations? Would the Hoppe paper be good here?

Surprisingly, there has been little or no cross-pollination between these groups. This paper proposes a software- and hardware-agnostic production pipeline called Chimera which enables production houses in the film and broadcast animation industry to manage and render massive data sets using a single off-the-shelf computer.

Chimera applies contemporary real-time rendering techniques to traditional film and broadcast animation production. It can be implemented on setups as small as one off-the-shelf computer, though it is flexible enough to easily be expanded to larger production environments. The techniques combined to form this pipeline, along with the rationales for doing so, are described in Section 4.

A metric for comparing rendering techniques was devised and used to measure the visual quality and performance of Chimera. The experimental design and analysis of this metric is described in Section 5.

Discussion of the benefits of Chimera versus other rendering techniques can be found in Section 6.

Comment [BH3]: Don't forget to label sections in here. "Section X"

2. Key Terms

A **massive data set** is one that is so large and complex that existing methodologies, tools, and technologies can't cope with it readily (Ket-

Comment [BH4]: Citation? I like your definition though. Maybe say

We define a massive dataset as a set of graphics so large and complex…with it readily.

tenring, 1998). A **manageable data set** is one just beneath the threshold of the massive data set. If a 3,500,000 polygon count is determined to be massive, then 3,499,999 is its manageable equivalent.

Off-the-shelf, for the purposes of this paper, refers to software and hardware than can be purchased or acquired freely without extensive customization. Proprietary software or hardware is not included.

A **production pipeline** can either refer to the software and hardware processes and techniques used to render computer graphics or the various theories of professional role delegation when creating computer graphics in a team environment. This paper refers to the former definition.

Real-time rendering refers to interactive computer graphics outputted immediately to a viewing device, while **run-time rendering** refers to the process of computing a 3D scene in animation software and storing the results on hardware or media for later processing or viewing.

3. Problems

Compared to large visual effects companies, small- to mid-sized production houses in the film and broadcast animation industry face very different kinds of problems. Specifically, smaller production houses must get by with substantially fewer computing resources and manpower than their larger, better-equipped competitors. With respect to production pipelines, small production houses deal with at least three kinds of issues: hardware costs, model complexity, and pipeline secrecy.

3.1 Hardware Costs

By definition, massive data sets require immense computing power to be adequately managed. Large production houses address this problem with server farms or hundreds of networked workstations. For example, Weta Digital, Ltd., a Wellington, New Zealand-based production house, built a server farm of 3,200 processors to produce the visual effects seen in the Lord of the Rings films (Mitchell, 2004). Industrial Light & Magic's new facility in San Francisco, CA boasts over 3,000 networked processors (Scanlon, 2005). The cost of this equipment, not including setup, maintenance, and customization, numbers in the millions of dollars. Smaller production houses lack this budget, and must rely on other strategies to render their animations. To cut costs, some small companies have outsourced their rendering needs to brokers like Hewlett-Packard's utility rendering service ("HP and Alias," 2004) and IBM, through RenderRocket. The cost of these services is typically calculated per CPU-hour. When individual frames of an animation can require 90 CPU-hours or more to render, the 50- to 60-cent cost per CPU-hour quickly adds up (Borland, 2005).

Comment [BH5]: Citation…tired of me saying that yet? ☺

Alternately, small production houses may heavily rely on compositing to work around massive data sets. One type of compositing involves the process of rendering multiple elements of a sequence separately as layers and combining the layers into a finished frame. The contents of these layers may range from digital background plates to filmed miniatures. Compositors may also create the illusion of continuity by "stitching" together many short clips into one long sequence. Because each clip is rendered separately, a kind of digital "sleight-of-hand" must be employed to disguise the transitions between short clips and make the long sequence look uninterrupted. For example, a Powers of Ten animation which begins in outer space and zooms into a building on Earth's surface may use clouds and other distractions to hide the changes from shot to shot. If not executed properly, this process may lack the elegance and natural appearance of a truly continuous animation. Moreover, these compositing tricks force small companies, already heavily restricted by their budgets, to add compositors, compositing tools, and additional steps to an already complex pipeline. With the cost of real-time compositing solutions beginning at $100,000 per unit ("Discreet Delivers," 2003), it is in small production houses' best interest to reduce compositing needs whenever possible.

Comment [BH6]: I don't know if the animation would be considered simple

3.2 Model Complexity

The off-the-shelf software and hardware used by small production houses typically cannot handle massive data sets with many complex models. Preliminary tests for this paper revealed that an off-the-shelf computer running Alias Maya 7 can load about three million polygons into a scene before the program crashes. For modelers or animators to effectively manipulate the data for any task besides rendering requires even less complex scenes than this. Thus, artists are limited by their software and hardware in the

Comment [BH7]: Our tests…

amount of detail they can add to models and textures. Yet, expectations of quality remain high, as small production houses often produce special effects for such widely-seen venues as television commercials and films (Dolbier & Megler, 2005). “Audiences who have seen Pixar's best work, many of whom have logged hundreds of hours inside beautifully rendered video game worlds, are brutally critical of shortcuts” (Borland, 2005). To meet the demand for realism and detail, these companies must settle for lower resolution textures and models that don’t look as good, or more often, compositing tricks which increase pipeline complexity at the expense of visual continuity.

3.3 Pipeline Secrecy

Reliable, publicly-available information on animation production pipelines is extremely limited. The dearth can be attributed to intellectual property claims within the animation industry. With several large companies releasing animated features each year and countless smaller production houses competing for contracts, all specific processes and techniques are considered competitive advantages. No books have been written about animation production pipelines, though snippets of specific processes can be garnered from “making of” featurettes found on some DVDs. In addition, there seems to be no “industry standard” or generic production pipeline; experts familiar with the industry claim that every company does things their own way (Bettis, 2005). As a result, information on production pipelines can typically be gleaned only from non-academic sources such as company press releases, news articles, and trade magazines.

4. Chimera

Chimera is a publicly-available hybrid production pipeline designed to primarily benefit smaller animation houses which lack supercomputers or render farms and extensively customized proprietary software. To address these specific needs, Chimera is software- and hardware-agnostic, though for the purposes of this research it has been implemented on an off-the-shelf desktop computer running Alias Maya 7. It consists of two main components working in concert: normal mapping and level of detail. By marrying techniques typically associated with real-time rendering with traditional techniques for animation production, Chimera enables smaller production houses to produce animations using massive data sets.

Chimera allows for two possible production workflows, depending on the needs of its user (see Appendix A). One option is to work from high polygon models, such as laser-scanned models, and simplify the models incrementally using a progressive mesh or CLOD algorithm. The other option is to work from a simple base mesh and continually add detail to the model until a sufficient level of complexity is reached. Once the models are created, UV unwrapping must occur to ensure that the models’ textures are comparable. Normal maps can then be created from the high polygon models and mapped onto the low polygon models. Once the low polygon models are mapped they can be placed in the scene. The user can now animate and prepare the scene for rendering. The last step before render is to set up the level of detail script. The user can then render the scene.

The benefits of normal mapping and level of detail are briefly explained in Sections 4.1 and 4.2, respectively. Section 4.3 describes the reasons for combining the two techniques, and Section 4.4 outlines the development process for Chimera.

4.1 Normal Mapping

Normal mapping is a type of bump mapping first introduced at SIGGRAPH 1996 (Krishnamurthy & Levoy, 1996). One application of normal mapping permits 3D models with low polygon counts to appear more detailed and complex by instructing the renderer to replace the model’s surface normals, resulting in more detailed shading (Cignoni, Montani, Rocchini, & Scopigno, 1998). The normal map itself is a 2-dimensional texture with normal information encoded in the red, green, and blue (RGB) color channels of the image. Animators can generate a normal map from a high polygon model and apply it like any other texture to a low polygon model. The calculations required for producing the normal information are “baked” into the normal map so the renderer needs not calculate them again at render time. A normal-mapped low polygon model can appear highly detailed but the actual polygon count and rendering time can actually be decreased. Dan Prochazka, Product Manager for 3D Animation Software at Discreet, echoes these advantages:

Comment [BH8]: Citation

Comment [BH10]: If you’re describing a normal map here, can you delete it from the terms page? I think you could get rid of the definitions section to save pages.

Comment [BH11]: Citations?

Comment [BH9]: I thought there was three?

Maybe it’s “…normal mapping and level of detail implemented through a run-time script we developed.”

You don’t have to have a section on the script…just include a few sentences in with level of detail

> The primary benefits of normal maps are time-savings in rendering and the ability to fit more objects into a scene because everything can be at a much lower resolution. So, you don't have to worry if your pipeline can't deal with massive data sets — with normal maps the eventual output is the same as it would be if everything in-scene was at an incredibly high resolution. (Moldstad, 2004, para. 3)

Normal mapping has been embraced for interactive and real-time computer graphics applications, particularly gaming and virtual reality, because normal mapping permits highly detailed images to be generated from very low polygon models. In real-time rendering, visual quality often comes second to frame rate and performance; 15 to 30 frames per second are required for acceptable interactive navigation (Constantinescu, 2000). However, normal mapping has not received equivalent attention from the film and broadcast animation industry because off-the-shelf animation packages did not come equipped with tools to easily generate normal maps until very recently. The newest versions of industry-standard animation packages such as Discreet 3D Studio Max, Alias Maya, and SoftImage XSI have been furnished with normal mapping capabilities in response to, and resulting in, increased interest in the benefits of normal mapping technology. Pixelogic zBrush, a 3D modeling package, has popularized normal mapping by generating normal maps from extremely complex models created in the software and exporting lower polygon models that look comparable.

4.2 Level of Detail

Level of detail (LOD), another concept borrowed from real-time and interactive computer graphics, has been employed since the 1970s (Heok & Daman, 2004). LOD refers to the process of swapping among several versions of the same base model depending on predetermined importance criteria. One application of LOD relates to a model's distance from the virtual camera in a 3D scene. For example, a low resolution version of the model is displayed when the model and camera are far away from each other, and higher-resolution versions of the model are displayed as the distance between the model and the camera decreases. Other techniques for LOD selection include size, eccentricity, depth of field, velocity, fixed frame rate, and culling (Constantinescu, 2000). Computing performance can be improved by using low detail models in a scene when their details are unlikely or impossible to be perceived by the human viewer.

Two level of detail frameworks include continuous level of detail (CLOD) and view-dependent level of detail (Heok & Daman, 2004). CLOD employs an algorithm to automatically reduce polygons via a multiresolution or "progressive" mesh (Hoppe, 1996). In some user-controlled manifestations, this process benefits from being extremely easy to use; an animator need only specify the starting and ending polygon resolutions and the CLOD algorithm takes care of the rest. The result boasts superior fidelity; however, the computation required to interpolate polygon resolution at each frame is expensive.

While a variety of CLOD algorithms have been written, few have been ported from an academic context to work with off-the-shelf animation packages. In addition, industry professionals have been hesitant to adopt academic LOD algorithms. At the 2003 Game Developers Conference in San Jose, CA, Robert Huebner of Nihilistic Software described some of the biggest problems of integrating academic LOD algorithms into his company's pipeline. Most studies pay little attention to the effects of vertex shading, texture, and UV coordinates, which are priorities for game developers. He also found that most algorithms could not maintain data in a format that computer hardware could process directly, for maximum performance gains (Huebner, 2003).

An alternative to CLOD, view-dependent level of detail, works by swapping different versions of a model based on the virtual camera's "view." Human perception generally can not discern between a low detail and high detail model at a certain distance, though this threshold is dependent on many factors including model shape (geometric, organic, animal, man-made, etc.), model complexity (Watson, Friedman, & McGaffey, 2001), and the actual distance from the camera. The term "popping" is used to describe the undesirable situation where a model swap is noticed by the viewer, but with care popping can be minimized or altogether avoided.

Comment [BH12]: …guess what? Citation!

In real-time applications that use view-dependent level of detail, the rendering engine must "guess" when to swap models. This decision is usually based on a formula that considers model importance, proximity to the active camera in the scene, and other factors. The lack of human input in this process means that popping frequently occurs; thus, this kind of LOD by itself is unsuitable for most run-time rendering applications, such as film and broadcast animation production, where popping is not acceptable.

Fortunately, run-time applications, by definition, need not be rendered immediately and adjustments can be made by animators. Human-calibrated LOD is more time-consuming but can be fine-tuned to eliminate popping. Animators can choose how many levels of detail they wish to use as well as when each model can be swapped out. This is typically done by importing several versions of a model into a scene and keying the visibility of each version to allow each to appear and disappear at various points throughout the animation. The process is tedious, requiring substantial trial-and-error on the part of the animator, and the presence of several versions of each model in a scene exponentially complicates the process. Without scripting to aid the process, even human-calibrated LOD is impractical for run-time rendering scenes with advanced camera movement and more than a few different models.

4.3 Combining Normal Mapping and Level of Detail

While both normal mapping and LOD have been extensively proven to improve performance for real-time rendering, their adoption for run-time rendering for film and broadcast animation has been hindered by technological barriers. Until recently, normal mapping was difficult or impossible to implement at run-time using off-the-shelf animation packages. LOD, while adequate for interactive 3D animations where popping is acceptable, is too slow and clumsy to implement in run-time animations where human calibration is the only feasible option for achieving continuous sequences.

Normal mapping and LOD have been successfully married for interactive 3D applications. Hoppe produced one such interactive demo using progressive meshes based on his SIGGRAPH 2001 paper (Sander, Snyder, Gortler & Hoppe, 2001). Chimera transfers these ideas to the realm of run-time rendering for the purposes of managing massive data sets.

4.4 Development Process

The first step in developing Chimera was the acquisition of a sample massive data set. Stanford University's collection of 3D-scanned models proved to be an invaluable source of complex models appropriate for academic research. These models feature extremely high detail—many with hundreds of thousands of polygons—and a variety of intricate surfaces. They are free for academic use, enjoy a long history of use in computer graphics research, and quickly provided this research with a variety of model types.

After examining many professional animation packages, Alias Maya 7 was selected as the off-the-shelf animation package to be used for this paper primarily because of its ubiquity in industry, especially at smaller companies that can't afford to write their own animation software (Goldman, 2001). "Maya is the absolute, undisputed industry-standard in 3D animation software," said Adam Yaniv of Rhythm & Hues, a Los Angeles, CA-based production house ("Vancouver," 2005, para. 7). Maya enjoys a large user community with plenty of support and interest. More practically, researchers in this study were most familiar with Maya as compared to other packages and could develop in the software most easily.

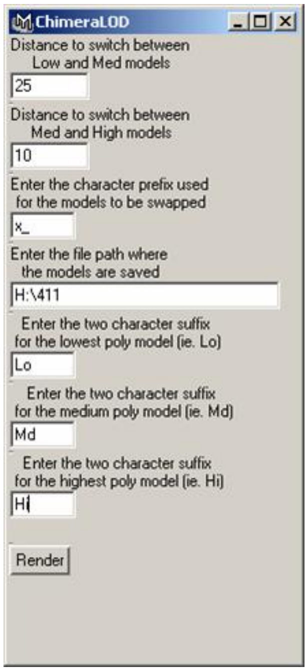
A workstation in Purdue University's Department of Computer Graphics Technology served as an off-the-shelf computer for this paper. The workstation, a Dell Optiplex GX280, includes a 3.2 GHz Pentium 4 processor, 1 GB of RAM, dual 160 GB SATA hard drives, and an nVidia Quatro 7800 video card.

Erik Pojar's Progressive Mesh plug-in was initially selected for Chimera's LOD computation (Pojar, n.d.). One of the only open-source, freely available CLOD plug-ins for Maya, the Progressive Mesh plug-in is derived from the QSlim algorithm (Garland & Heckbert, 1997), whose quadric error metric (QEM) is one of the best available for baseline simplification (Heok & Daman, 2004). It features a straightforward GUI and allows animators to "paint" on each progressive mesh which areas should be priorities for retaining detail and complexity. While the Progressive Mesh plug-in performed well on individual frames, it was soon discovered that an off-the-shelf computer could not render animated sequences of frames with this algorithm without crashing. The calculations required for polygon reduction from frame to frame was simply too processor-intensive.

The Progressive Mesh plug-in was not completely abandoned for this paper; surprisingly, it provided functionality that greatly assisted the efficiency of normal mapping portion of the Chimera pipeline. Chimera permits modelers to adopt one of two possible modeling techniques. A modeler may begin with a low polygon model, preserve this version, and add details until a high polygon model is created. Alternately, a modeler may begin with a high polygon model and use a CLOD algorithm to produce a low polygon model. In this case, the Progressive Mesh plug-in allowed modelers to start with a high polygon model and easily create additional versions of that model at any specified level of detail, simply by adjusting a "resolution" slider in the plug-in. Since all models are derived from the same base model using this technique, their UV coordinates remain aligned. Modelers can then use these models to create normal maps. This process is a substantial time-saver over manually reducing polygons, or creating a new low polygon version from scratch and attempting to align UV coordinates by hand.

Since the Progressive Mesh plug-in was too computationally taxing for the LOD portion of the Chimera pipeline, a more efficient method was necessary. View-dependent LOD frameworks change model complexity less often and are therefore less processor-intensive. No user-controllable view-dependent LOD plug-ins for Maya could be found, so one was developed from scratch in the form of a MEL script (see Figure 1).

Comment [BH13]: It's a script and not a plug-in. (I changed them all for you)



Distance to switch between Low and Med models: 25

Distance to switch between Med and High models: 10

Enter the character prefix used for the models to be swapped: x_

Enter the file path where the models are saved: H:\411

Enter the two character suffix for the lowest poly model (ie. Lo): Lo

Enter the two character suffix for the medium poly model (ie. Md): Md

Enter the two character suffix for the highest poly model (ie. Hi): Hi

Figure 1. MEL script GUI.

The MEL script dynamically swaps models and textures at run-time at user-specified distances. To address the issue of multiple versions of each model in the scene consuming excessive resources, versions that are replaced are unloaded from memory. While this implementation is coarser than a CLOD algorithm, it adequately reduces the massive data set by only loading complex models into memory when viewers will appreciate the detail. A GUI within Maya allows the user to input camera distances for each level of detail transition, as well as prefixes and suffixes to associate different versions of each model.

The script must be fine-tuned or "calibrated" by the user to ensure models switch at the proper threshold; that is, soon enough that there is no popping, but late enough that computer performance is improved by rendering low-detail models whenever feasible. Human calibration is an unfortunate necessity until research reveals a perfect formula or ideal threshold for calculating level of detail transitions (Pajarola & Rossignac, 2000).

5. Metrics

Changes in rendering techniques can dramatically affect the visual quality and performance of an animation, so both variables must be measured to draw meaningful conclusions. Two visual tests and a performance test were designed to measure how Chimera compares to other rendering techniques.

Many possibilities exist for measuring visual quality. The visual tests in this paper measure similarity, which is “imperative” for applications like movie special effects (Lindstrom, 2000, p. 103), in terms of perceived detail. Because film and broadcast animations are produced primarily for aesthetic reasons, human visual perception of similarity is a more appropriate metric than computed comparisons with tools like Metro and MeshDev, even though perceived similarity is more difficult to accurately measure (Luebke, 2001; Watson, Friedman, & McGaffey, 2000). For the purposes of this paper, what people perceive as looking similar is more important than what is geometrically similar.

5.1 Visual Test 1 (Ranking)

Purpose. The purpose of the ranking test was to compare animations rendered with Chimera, normal mapping, LOD, or traditional high polygon models in terms of similarity. The hypothesis was held that participants would find normal mapping and Chimera to look the same, while LOD would look less detailed. The results of this paper would then permit a comparison of normal mapping and Chimera’s performance. The experiment design is based on visual fidelity preference measurements for LOD still images (Watson, Friedman, & McGaffey, 2001), but the addition of a time variable dramatically altered the requirements of the instrument design. Greg Francis of Purdue University’s Department of Psychology was consulted to validate the changes with respect to the experiment’s goals.

Method. Twenty three undergraduate students, mostly freshmen, participated in the proctored experiment. All of the students had some familiarity with computer graphics and basic knowledge of 3D animation practices. An unproctored, identical version of the experiment was made available on the Web to collect the maximum number of responses, which were categorized separately. In total, 778 proctored and 1,615 unproctored data points were recorded.

Participants were shown two still images juxtaposed on a computer screen in a forced dichotomy scenario. They were asked to use the mouse to click on the image which seemed to have more detail. The computer recorded which image was chosen (the “winner”) and which was not chosen in a database. The participant would then be shown another comparison and asked to perform the same task again. Proctored participants were asked to make at least twenty comparisons; unproctored participants could make as many or as few as they liked. To avoid prejudicing the results, participants were not told what the test measured or what each image represented.

The images shown to the participants were highly varied to acquire a breadth of data points. Three 10-second animations were used, featuring three different models—the bunny, Buddha, and dragon from Stanford University’s repository. Each of the three animations was rendered using Chimera, normal mapping, LOD, and high polygon models for a total of twelve animations. For this test, Chimera and LOD swapped models twice: from 0.5% to 2% polygon resolution at frame 75, and 2% to 5% polygon resolution midway at frame 150. The normal mapped models maintained 5% polygon resolution throughout the animations. A standard template was used for all the animations: the virtual camera in the scene would be positioned far away from a single model so that it could barely be seen; the camera would then zoom in towards the model until it nearly filled the entire frame; the camera would then rotate 360° around the model to capture a variety of angles. To avoid confounding variables, the models were untextured except for normal maps (where applicable) and the scenes were solid black with default lighting. Participants were shown a randomized frame (1 through 300) of one of the models rendered with two randomized techniques which were always different from each other. The model type was also randomized for each comparison.

Results. Figure 2 depicts the results of the ranking test. Each rendering technique’s comparative win percentage (i.e., wins vs. total comparisons) is displayed on the y-axis. Proctored and unproctored results were calculated separately and averaged.

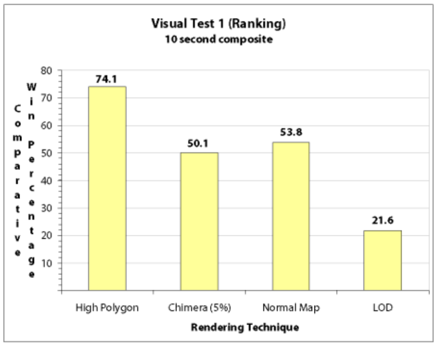


Figure 2. Ranking test results.

Statistical analysis showed that no significant difference exists between the animations rendered with Chimera and normal mapping, regardless of camera position. In addition, animations rendered with LOD lost comparisons to the other techniques almost all of the time. These results confirm the stated hypothesis that normal mapping and Chimera look the same, but it remained unclear how these techniques compared to high polygon models. In this test, high polygon models won comparisons to the other techniques at close frames (greater than 150) almost three-fourths of the time. However, the high polygon models were composed of 95% more polygons than the other techniques. Before performance data would be relevant, another test was necessary to determine if Chimera (and its statistically similar partner, normal mapping) could be made visually comparable to high polygon models.

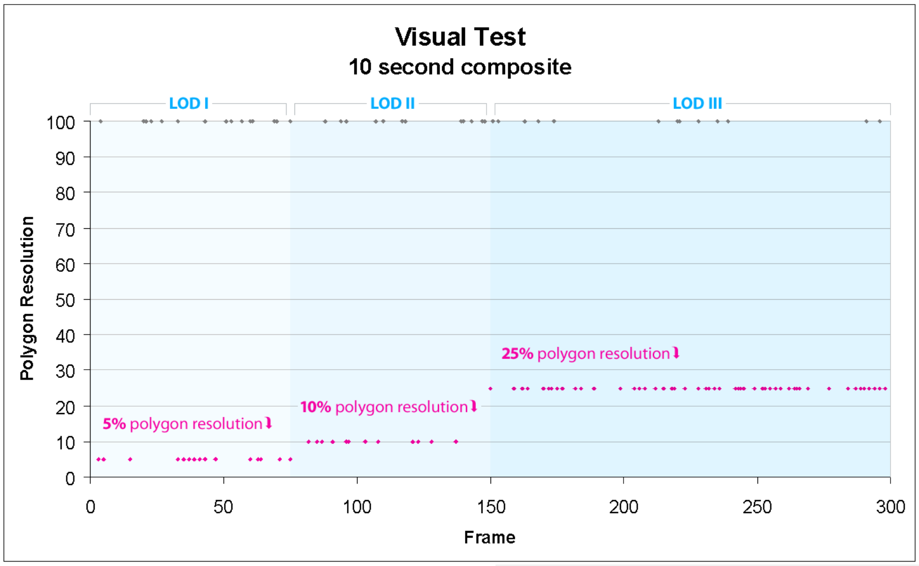
5.2 Visual Test 2 (Preference)

Purpose. The purpose of the preference test was to compare animations rendered with Chimera at different levels of detail to traditional high polygon models. The ranking test revealed that high polygon models looked more detailed than Chimera models at close frames when Chimera's polygon resolution was 5% of the high polygon models. However, it was possible that the two techniques would be visually comparable when Chimera's polygon resolution was higher than 5%. The hypothesis was held that at some polygon resolution greater than 5%, Chimera would be visually comparable to high polygon models.

Method. This experiment was conducted via the Web to collect the maximum number of responses, with participants solicited from Purdue University's Computer Graphics Technology listserv. Most subscribers to this listserv are faculty, alumni and current students of Purdue University's Department of Computer Graphics Technology, so it can be assumed that many of the participants were familiar with computer graphics and had basic knowledge of 3D animation practices. In total, 645 data points were collected.

The testing procedure of the preference test was identical to that of the ranking test, with a few important exceptions. In this experiment, Chimera was visually compared to high polygon without level of detail transitions. This setup permitted an analysis of Chimera at a constant level of detail throughout each animation, thereby identifying a time threshold for when (if ever) Chimera surpassed high polygon models. Chimera animations were rendered at polygon resolutions of 5%, 10%, 25%, and 50% of the high polygon models. The 5% polygon resolution animations served as the control in this experiment; the ranking test revealed that participants would choose high polygon models over Chimera at 5% polygon resolution almost all the time. Animations rendered with normal mapping or LOD were not included in the preference test.

Results. Figure 3 illustrates the detailed results of the preference test. Each point represents a "win" for that particular polygon resolution compared to one of the others (the comparison always pits a high polygon "control" against a version of Chimera). Thus, more points at a given polygon resolution over a particular time frame



indicate the participants’ preferred polygon resolution for that time frame. The graph is divided into three regions representing three level-of-detail changes (5%, 10%, and 25%). Relationships between polygon resolution (on the y-axis) and time (on the x-axis) indicate how camera distance affects a participant’s ability to discern one technique from another. The results reflect a composite of wins for all three model types. Figure 4 presents the same results with the time variable omitted.

Figure 3. Detailed preference test results.

Results of the preference test revealed that at no point in the animations did participants consistently prefer high polygon models over Chimera. At times when the virtual camera in the 3D scene was far away from the model, participants chose either technique about half the time. As the camera moved closer to the model and participants could perceive more details, Chimera was preferred more often and high polygon models were preferred less often. At very close frames (greater than 250), Chimera won comparisons to high polygon models almost all the time. The difference in preference was most drastic when Chimera’s polygon resolution doubled from 5% to 10% of the high polygon models, and in fact was not much different from the results at resolutions of 25% and 50% of the high polygon models. Resolution, rather than camera distance, played a greater role in changing preferences from high polygon models to Chimera. With Chimera established as equivalent or superior to high polygon models in terms of perceived detail, a performance test comparing the two techniques in terms of rendering time could be conducted.

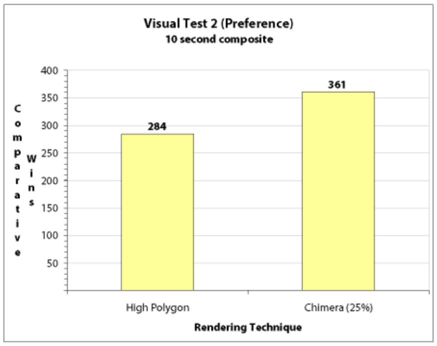


Figure 4. Preference test results sans time.

5.3 Performance Test

Purpose. The purpose of the performance test was to compare each of the four rendering techniques—Chimera, normal mapping, LOD, and high polygon models—in terms of their rendering time, or how much computational processing was required to produce a sequence of frames for a given animation. The starting point for the animation in the performance test is a massive data set in the form of a highly complex scene with multiple models and extensive camera work. The hypothesis was held that Chimera would require the least rendering time, with LOD, normal mapping, and high polygon models requiring incrementally more rendering time.

Method. A 3D scene consisting of a massive data set was constructed in Maya to be used in the performance test. To create a massive data set, high polygon models were imported into the scene until Maya crashed. Then the scene was recreated with one less model than the number which caused Maya to crash. In Maya using an off-the-shelf computer, a massive data set which crashed the program was determined to be about 3.25 million polygons, while a manageable data set, meaning the maximum number of polygons Maya could handle without crashing, was defined as one less model than the massive data set, approximately 3 million polygons. A virtual camera was set up in the scene similar to that of the visual tests. Models in the scene were dispersed so that some would be directly in the line of sight of the camera while others would be located peripherally, behind, or in front of the camera’s focus. The goal of this setup was to simulate a practical situation in broadcast or film animation where a massive data set would be used; for example, a camera zooming into a group of figures or vehicles moving across a landscape.

The scene was rendered using each of the four rendering techniques, with maximum polygon resolution capped at 25% of the high polygon models, as with the preference test. For consistency, the animation was limited to 10 seconds (300 frames) in length. For this test, Chimera and LOD again swapped models twice: from 5 to 10% polygon resolution at frame 75, and 10% to 25% polygon resolution midway at frame 150. Performance was measured in terms of rendering time; i.e., how much time it took to completely render all 300 frames of the animation from start to finish.

Results. Results of the performance test are displayed in Figure 5. Normal mapping required the least amount of rendering time, while high polygon models required the most. The hypothesis that Chimera would perform the best was rejected. Some useful generalizations of these results can be made. In terms of rendering time in highly complex scenes with multiple models and extensive camera movement, Chimera, normal mapping and LOD dramatically outperform high polygon models. Furthermore, the performance of Chimera, normal mapping, LOD performance is clustered at around 6 minutes, whereas, high polygon performance is much worse at over 50 minutes. In summary, the high polygon models rendered much slower than any of the other techniques.

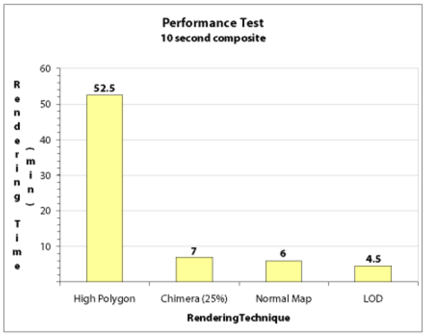


Figure 5. Performance test results.

6. Discussion

6.1 Surprises

The results of both the visual tests and the performance test were surprising. The ranking test set the stage for the hypothesis that high polygon models would always look more detailed than other techniques, yet the preference test revealed that this is not always the case. The preference test showed that when the distance between the virtual camera in a 3D scene and the model is large, participants can not discern between high polygon models and Chimera at any level of detail. This is expected; when models look very small, it is difficult for the human eye to detect minute differences. However, as the distance between the camera and the model decreased, Chimera not only caught up to high polygon models in terms of comparative wins, it actually surpassed it by a significant margin. Equivalent preference between Chimera and high polygon models would seem to make sense because the Chimera model has much less actual detail and complexity than high polygon models; at close frames, Chimera had 75% fewer polygons. It seems that normal mapping more than compensated for the decreased resolution. While human perception is the metric for measuring similarity in this paper, it is probable that Chimera seemed more detailed at close frames because normal maps can exaggerate minor topographical variations. Further exploration into this area may uncover additional explanations.

The rendering times of each technique in the performance test were also unexpected. It was hoped that Chimera would outperform the other techniques in rendering time because it rendered fewer polygons than high polygon models or normal mapping when the distance between the virtual camera in the 3D scene and the model was large. However, an unexpected complication arose in the design of the MEL script responsible for swapping the models in animations rendered with Chimera and LOD. When models are swapped at each level of detail, the computer must load the new model into memory, and this may require additional time, depending on the complexity of the model and the number of swaps occurring at a given frame. For this reason, the MEL script allows more models to be initially loaded into a scene, but at the expense of some rendering time. For the performance test in this paper, the increase of rendering time was just 37%, as compared with 1,084% for high polygon models. However, it is not known at this time how rendering time may be affected by longer animations, more complex scenes, or different kinds of camera movement. More performance tests must be conducted to understand the implications of these differences.

6.2 Recommendations

In certain situations, Chimera is the only viable technique for rendering a 3D scene, while in others, several may work equally well or better. A number of factors must be considered when evaluating which technique to use. This section provides guidelines for when normal mapping, Chimera, LOD, and high polygon models are appropriate choices.

Normal mapping

- Normal mapping presents the best combination of visual quality and performance.
- Normal mapping alone won't work for massive data sets; however, it is a good baseline technique to use when Chimera is not necessary.
- Avoid normal mapping when the camera moves very close to models and low polygon counts may be noticed.
- Use normal mapping for scenes which require high detail and a moderate number of models, or moderate detail and a large number of models.
- Use normal mapping for scenes which require minimal camera movement.

Chimera

- Chimera takes slightly longer to render, but allows more models to be initially loaded into the scene.
- Use Chimera for scenes which require high detail and a large number of models.
- Use Chimera for scenes which require extensive camera movement, particularly camera zooms that take advantage of LOD.

Level of detail

- LOD offers the worst visual quality but the second-best performance.
- Use LOD for scenes which require low detail and a large number of models.
- Low detail scenes are seldom desired for broadcast animation, so LOD by itself is unlikely to be a best choice for most run-time rendering scenarios.

High polygon models

- While the visual quality of high polygon models is comparable to Chimera and normal mapping, it performs much worse than the other methods.
- Use high polygon models for scenes which require high detail and a very small number of models.
- Use high polygon models for scenes which require minimal camera movement.

Since Chimera and normal mapping look and perform so similarly, it's important to know when to choose one technique over the other. Ultimately, no technique seems to be universally appropriate. The decision of which rendering technique to use depends on the individual requirements of each scene; specifically, the number of models, the detail of those models, and the nature of the camera movement in the scene (see Figure 6). The best pipeline may make use of all of these techniques for different parts of an animation.

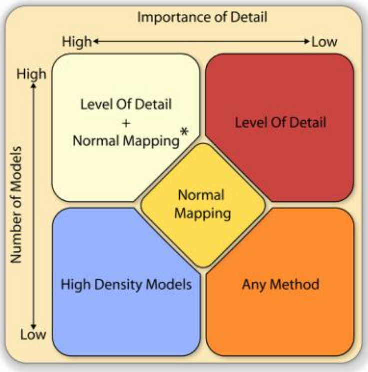


Figure 6. Pipeline usage guidelines.

It is also important to point out that pipeline managers will ideally know early on approximately how many models and how much camera movement a scene requires, in order to inform modelers if they need to create multiple versions of a model at different resolutions. However, changes to this pipeline can be made relatively cheaply and quickly during production. Since normal mapping and Chimera renders so much faster than high polygon models, the time spent creating these multiple versions will still likely be much faster than rendering only high polygon models from start to finish, because re-renders do not take nearly as long. As a result, this pipeline could result in a lower cost to produce an animation or it could create extra time to be used for revisions or improvements to an animation.

7. Conclusions

Managing and rendering massive data sets presents considerable challenges to small- and mid-sized production houses in the film and broadcast animation industry. Barriers such as hardware cost, model complexity, and pipeline secrecy pose special problems for these smaller companies to produce high quality work without proprietary software or expensive hardware solutions.

Chimera, a hybrid production pipeline which applies interactive 3D techniques to traditional animation production, enables smaller production houses to manage massive data sets using off-the-shelf software and hardware of their choice. Normal mapping and LOD, two techniques borrowed from gaming and real-time visualization, offer time-saving benefits to film and broadcast animation and run-time rendering as well. When combined, they permit animators to fit more highly detailed models into a scene than other rendering techniques.

Experimental data shows that Chimera and normal mapped animations can look as detailed as high polygon models. In addition, Chimera and normal mapped animations perform better than high polygon models in terms of rendering time. Chimera offers a unique solution to specific problems in film and broadcast animation; however, its advantages and disadvantages must be compared to normal mapping and other rendering techniques. The best pipeline depends on the individual requirements of a scene and combines different techniques to achieve the best results.

8. Further Work

The development of a novel production pipeline sets the stage for a great deal of future research. The effect of textured versus un-textured normal-mapped models on detail perception would answer important questions about why people find normal mapped models more detailed than high polygon models. An effective solution for rendering scenes with primarily geometric, rather than organic, models would neatly supplement this paper. Additionally, it is currently unknown how the pipeline described in this paper would perform with manually-created models, as only laser-scanned models have been tested thus far. Implementation of mipmapping, external memory management and out-of-core simplification may improve the pipeline's performance. Finally, this paper used detail as an indicator of similarity for comparing models rendered with different techniques. Other metrics for similarity exist, and the value of this pipeline as measured by other metrics may answer important questions about what truly matters with respect to human visual perception.

Acknowledgements

Thanks to Ben Watson of Northwestern University for the information he provided about his visual fidelity research. Greg Francis of Purdue University gave useful feedback on designing visual perception preference experiments. Ben Tyner of Purdue University's Statistical Consulting Service offered invaluable assistance with data analysis. Erik Pojak's Progressive Mesh plug-in enabled a high-to-low density modeling process. Bryan Ewert's MEL scripting resources were greatly appreciated. Gratitude must be directed towards the Stanford Computer Graphics Laboratory for the use of its laser scanned models of the bunny, Buddha and dragon. Bedrich Benes of Purdue University kept this research project pointed in the right direction from start to finish. Finally, Carlos Morales of Purdue University contributed his support and expertise.

References

Bettis, D. (2005). Digital production pipelines: examining structures and methods in the computer effects industry. Unpublished master's thesis, Texas A&M University.

Borland, J. (2005, August 15). 'Shrek' producer's pigeon play. CNET News.com. Retrieved December 12, 2005, from http://netscape.com.com/Shrek+producers+pigeon+play/2100-1026_3-5830819.html.

Cignoni, P., Montani, C., Rocchini, C., & Scopigno, R. (1998). A general method for preserving attribute values on simplified meshes. In Proceedings of IEEE Visualization 1998, 59-66.

Cohen, P. (2005, January 31). Alias announces Maya 6.5. MacWorld. Retrieved December 12, 2005, from http://www.macworld.com/news/2005/01/31/maya65/index.php.

Constantinescu, Z. (2000). Levels of detail: An overview. In Nonlinear Analysis: Modelling and Control (5), 39–52.

Comment [BH14]: Good work Kurt, I really liked it. It was easy to follow and well thought out. I think you could keep only one part of the definitions…you don't need to define terms twice. On the 411 site should be the specifications for the font size, type and margin and column widths, etc. Good luck Call me if you need anything else. For the references I would look at our previous research. I will send along our annotated Bibliography for you in case you don't have it.

The blue word is just one I added in.

Coral, C., Torrance, K., & Greenberg, D. (1984). Modeling the interaction of light between diffuse surfaces. Proc. SIGGRAPH 84. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 213-222.

Discreet delivers inferno 5, flame 8 and flint 8. (2003, January 23). Autodesk Press Room. Retrieved December 12, 2005, from http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=2482433&linkID=1977925.

Dolbier, G. & Megler, V. (2005, July 20). Understanding the existing infrastructure. Build a digital animation system. Retrieved December 12, 2005, from http://www-128.ibm.com/developerworks/library/wa-animstudio1/.

Garland, M. & Heckbert, P.S. (1997). Surface simplification using quadric error metrics. Proc. SIGGRAPH 1997. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 209-216.

Goldman, M. (2001, May 5). Building character. Animation Artist. Retrieved December 12, 2005, from http://www.animationartist.com/2001/05_may/features/building1.htm.

Heok, T.K. & Daman, D. (2004). A review on level of detail. Proc. International Conference on Computer Graphics, Imaging and Visualization (CGIV) 2004.

Hoppe, H. (1996). Progressive meshes. In SIGGRAPH 1996, 99-108.

HP and Alias sponsor multi-user animation showcase to highlight benefits of utility computing. (2004, September 22). Retrieved December 12, 2005, from http://www.hpl.hp.com/SE3D/se3d-release.html.

Huebner, R. (2003). Level of detail for games [Course notes]. San Jose, California: 2003 Game Developers Conference. Retrieved December 12, 2005, from http://lodbook.com/course/2003/Huebner_Games.ppt.

Kettenring, J. (1998). Massive data sets, data mining, and cluster analysis [Course notes]. Twin Cities, MN: University of Minnesota. Retrieved December 12, 2005, from http://www.ima.umn.edu/kdi/kettenring/sld001.html.

Krishnamurthy, V. & Levoy, M. (1996, August). Fitting smooth surfaces to dense polygon meshes. Proc. SIGGRAPH 96. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 313–324.

Lindstrom, P. (2000, November). Model simplification using image and geometry-based metrics. Ph.D. dissertation, Georgia Institute of Technology. Retrieved December 12, 2005, from http://www.cc.gatech.edu/~lindstro/papers/dissertation/dissertation.pdf.

Luebke, D. (2001). A developer's survey of polygonal simplification algorithms. IEEE Computer Graphics and Applications.

Mitchell, R.L. (2004, October 4). Blades, camera, action! Computerworld. Retrieved December 12, 2005, from http://www.computerworld.com/hardwaretopics/hardware/server/story/0,10801,96284,00.html.

Moldstad, F. (2004). Hidden gems of animation: Features that deserve more attention. Film Imaging. Retrieved December 12, 2005, from http://www.filmimaging.com/articles/viewarticle.jsp?id=27296&afterinter=true.

Pajarola, R. & Rossignac, J. (2000). Compressed progressive meshes. IEEE Transactions on Visualization and Computer Graphics, 11, 79-93.

Pojar, E. (n.d.). User controlled creation of progressive meshes. Retrieved December 12, 2005, from http://www.pojar.net/ProgressiveMesh/index.html.

Sander, P., Snyder, J., Gortler, S., & Hoppe, H. (2001). Texture mapping progressive meshes. Proc. SIGGRAPH 2001. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 409–416.

Scanlon, J. (2005, May). The new heart of the empire. Wired, 13, 5, 146-149.

Shaffer, E. & Garland, M. (2005). A multiresolution representation for massive meshes. IEEE Transactions on Visualization and Computer Graphics, 11, 139–148.

Vancouver Film School ups its Maya and MotionBuilder instruction. (2005, October 12). Vancouver Film School Press Room. Retrieved December 12, 2005, from http://www.vfs.com/thisisvfs.php?section=press_room&news_id=401.

Watson, B.A., Friedman, A., & McGaffey, A. (2000). Using naming time to evaluate quality predictors for model simplification. Proc. ACM Computer Human Interaction(CHI) 2001, 113-120.

Watson, B.A., Friedman, A., & McGaffey, A. (2001). Measuring and predicting visual fidelity. Proc. SIGGRAPH 2001. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 213-220.

Appendix A: Chimera Pipeline