

Motion and Scene Complexity for Streaming Video Games

Mark Claypool
Computer Science & Interactive Media and Game Development
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609, USA
claypool@cs.wpi.edu

ABSTRACT

This work provides detailed measurements of the motion and scene complexity for a wide variety of video games. Novel algorithms to measure motion and scene complexity are presented and then applied to over twenty-five games from a range of game genres. Direct comparisons of motion and scene complexity are made to traditional video, with differences highlighted in order to help thin client systems perform better when streaming video games. Preliminary performance measurements of current thin client systems are presented, evaluating the efficacy of streaming video games in the context of the motion and scene complexity measurements provided.

1. INTRODUCTION

The growth in connectivity and capacity of today's computer networks have enabled new forms of distributed computing where data and processing on a remote machine is acted upon on a local machine. This centralized model of computing allows a performance focus at a single location, the server, and enables user mobility and ubiquitous access for the clients. In fact, thin clients, where the local computer is primarily an input and output device and the remote computer does the majority of the processing, have seen a resurgence in use because today's networks can support bandwidth-intensive, client-server interactions.¹ Thin clients were made popular in the late 1980's by the use of X-terms, but network connectivity has made them increasingly viable today [14]. Computer games have seen tremendous growth in recent years, as well, with the end of 2006 seeing gross revenue from U.S. computer game sales at \$7.4 billion,² on par with domestic box office ticket sales.³ Networked games have also seen considerable growth, spurred on by the growth in residential broadband Internet connections with high capacities and low latencies that have encouraged game de-

¹In fact, this entire paper was written from a thin client connected to a server over a WAN.

²2006 Sales, Demographics and Usage Data, Entertainment Software Association, <http://www.theesa.com/>

³Total U.S. Box Office Grosses, National Organization of Theatre Owners, <http://www.natooonline.org/statisticsboxoffice.htm>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICFDG 2009 April 26-30, 2009, Orlando, FL, USA.
Copyright 2009 ACM 978-1-60558-437-9 ...\$5.00.

velopers to incorporate multi-player, networked features into their products. The convergence of these three technologies, networks, thin clients and computer games, provides a tremendous opportunity for sophisticated computing. Namely, a computer game hosted on a heavyweight, fat server being streamed as an interactive video over a network to be played on a lightweight, thin client. Use of thin clients is appealing for games under many circumstances: 1) when rendering a game scene that requires large amounts of data and specialized hardware that is not readily available at a client. For example, the Sony Play Station 3 (PS3) and Play Station Portable (PSP) support Remote Play [15], a thin client solution from the powerful PS3 to the less powerful PSP; 2) for augmented reality where a user interacts in the physical world but has game play enhanced through thin, wearable computers such as head-mounted displays connected to servers that control the game; and 3) personal computers that access a centralized game server where all players connect, easing maintenance of the game world and state consistency for multiplayer interactions.

Since computer games continually push the limits of hardware in an effort to provide more complex game world simulation and visual detail, efficient use of the network when streaming a video game is required in order to realize compelling game play. In particular, previous research [7, 13, 17] has shown that the content of a streaming video makes a difference when deciding on the best way to adapt video to limited capacities. Videos with low motion and low scene complexities have low bitrate requirements and do not need to reduce bitrates as much as do videos with high motion and high scene complexities. Moreover, videos with high motion and relatively simple scenes look better if all frames are kept but with low quality (also called *quality scaling*), while videos with low motion and complex scenes look better if some frames are discarded but those remaining have high quality (also called *temporal scaling*). Differences in the performance of quality scaling versus temporal scaling has been shown to correlate with video motion characteristics [17], and content-aware scaling has been shown to improve the perceived quality of streaming video by as much as fifty percent [13].

While there is a small set of videos traditionally used for multimedia research,⁴ precise measurements of their motion and scene complexity characteristics are not well-known. In fact, to the best of our knowledge, there are no established measures of motion and scene complexity that have been solidly correlated with user perception. Moreover, there is an even greater need for standard videos for game research. First, there needs to be a representative set of videos to be used as benchmarks for streaming research. Second, the motion and scene complexity characteristics of games needs to be studied, both to establish the benchmark and to ascertain the

⁴Videos such as Foreman, Paris, Coastguard, etc.

viability of streaming video games to thin clients.

In an in-depth effort to address some of these shortcomings, our research presents a suite of game videos, which were selected based on the perspective of the game player and carefully captured to obtain representative content. These game videos are provided as a benchmark for this, and future, research. Furthermore, this paper proposes novel measures of motion and scene complexity that correlate with user perception. These measures are applied to the selected games, providing the first of its kind analysis of the characteristics of games in relationship to the game perspective and in comparison with traditional videos. Lastly, preliminary experiments using a couple of popular thin client technologies and representative games begin to evaluate the efficacy of thin clients in supporting streaming video games.

This study seeks to answer the following questions (with a brief answer as revealed later in this paper provided in parentheses):

1. Can video encoding characteristics be used in an objective measure of motion and scene complexity correlated with user perception? (Answer: Yes, in the form of the percentage of forward/backward or intra-coded macroblocks (PFIM) and in form of the average of intra-coded block size (IBS).)
2. Do motion and scene complexity characteristics vary across computer games? (Answer: Yes, with camera perspective, first or third person, impacting both motion and scene complexity.)
3. Do motion and scene complexity characteristics differ for games and for video? (Answer: Yes, games vary more than typical videos in both amount of motion and scene complexity.)
4. Is video game streaming from fat servers to thin clients viable? (Answer: Yes, but only for games with low motion using a very low display resolution.)

The rest of this paper is organized as follows: Section 2 relates other work in measuring video characteristics and thin clients; Section 3 presents our novel measures of motion and scene complexity; Section 4 details game perspectives as they influence the effects of system settings; Section 5 describes our methodology to capture and analyze video traces of games; Section 6 analyzes the motion and scene complexity of the captured game videos; Section 7 provides preliminary evaluation of the performance of two thin client technologies; and Section 8 summarizes our conclusions and suggests possible future work.

2. RELATED WORK

This work overlaps research from two main areas: measuring of motion and scene complexity (Section 2.1) and measuring the performance of thin clients (Section 2.2). In addition, there is some related work specific to games and videos on thin clients (Section 2.3).

2.1 Measuring Motion and Scene Complexity

Peker and Dviakaran [12] present a framework for comparing the performance of motion activity with respect to a “ground truth” based on subjective ratings. They focus analysis on how effective MPEG motion vectors are in representing visual motion, showing that average motion vector length is not as effective as measures that discard some motion vectors (using the largest remaining) or that compute the variance of the motion vector magnitudes (as in MPEG-7).

Ma and Zhang [9] propose a shot-based measure of motion, namely the perceived motion energy spectrum (PMES) obtained by a temporal energy filter and a global motion filter. Their focus is on a measure that separates out object motion in the scene from cam-

era motion from, say, panning. Their brief evaluation shows their approach is effective when there are multiple objects moving in a video.

Tripathi and Claypool [13] propose a measure of motion based on the characteristics of the MPEG encoder. Their intent is to provide a quick method of informing a streaming server of the content characteristics in order to scale encoded video so as to maximize perceived quality. Evaluations of their system with user studies show content-aware scaling improves perceptual quality.

Neither the Ma nor Tripathi measure of motion is actually correlated with user-perception. While the Peker analysis is based on subjective user scores, a comparison with other metrics is not provided. Preliminary comparison of our proposed measure of motion suggests more effective representation of visual motion in a video as perceived by users than any of the above measures.

Lei and Bhattacharya [8] propose a measure of scene complexity based on geometries, such as intersections between curves and objects. Experimental results are given to illustrate the proposed method. Minu et al. [1] use scene based motion features as an element for robotic vision tracking. In particular, their experiments show the success of the approach to disambiguate features in complex scenes. Our proposed measure of scene complexity is simpler, being based on readily available MPEG statistics, even if the comparative correlation with user perception is not known at this time.

2.2 Measuring Thin Client Performance

Lai and Nieh [6] use a novel slow-motion benchmarking technique to evaluate the performance of several thin client platforms over Internet 2. They show general performance is often adequate for high capacity links, even when across the country. However, thin clients with more efficient bitrates can still have latency as the bottleneck to performance. They provide a summary of design choices that can aid thin client computing development for traditional applications.

Packard and Gettys [11] passively monitor network traffic between X clients and X servers under network capacity controlled conditions. A variety of test applications assess a fast understanding and quantification of performance issues. Among other findings, latency was found to dominate capacity in limiting performance for some applications and image transfers were found to dominate the overall network capacity used.

Billinghurst et al. [2] describe communication asymmetries for a wearable (and thin) display and a desktop computer. An accompanying user study is provided to determine the effects on collaboration, showing that the extent to which communication over a thin client is difficult largely depends upon the task being undertaken.

Kim et al. [5] describe pTHINC, a thin-client solution for PDAs that run applications such as Web browsers on more powerful, but remote, servers. With pTHINC, server-side scaling of the display provides improved performance for a variety of heterogeneous client displays. The results show pTHINC provides superior Web browsing performance for a PDA.

The above studies provide detailed insights into thin client performance, but predominantly pertain to traditional applications with no study specifically focusing on thin clients and games.

2.3 Games and Video on Thin Clients

Ocheltree et al. [10] propose a VESA standard for remote displays. They note short-comings with current thin client approaches for meeting the demands of video, as well as requiring customers to lock into a single operating system for client and server. The intent of the proposed standard is to enable clients to connect to heterogeneous servers and obtain performance comparable to that provided

by access to a local PC.

Winter et al. [16] propose a thin client system designed specifically for streaming and interactive games. Their system streams screen images after rendering by the graphics card, thus reducing bandwidth and increasing visual quality for streaming video games.

Lamberti and Sanna [7] present a framework for streaming video from a cluster of heavyweight PCs to thin clients, such as mobile phones. In their architecture, mobile clients remotely control a server cluster to encode frames generated by the graphics card into MPEG video and then streamed. Their evaluation shows complex 3D data sets can be displayed at up to 30 frames per second on heterogeneous thin client devices, such as PDAs and tablet PCs.

Our work complements these approaches by providing insight on the characteristics of streaming video games, enabling better scaling decisions for the streamed images after rendering, as well as evaluating two thin client technologies with several representative games.

3. MEASURING MOTION AND SCENE COMPLEXITY

A users study was conducted to subjectively measure the amount of motion and scene complexity for a range of videos, specifically the 9 video clips described in Section 5.3. The intent was to correlate these subjective measures with objective measures readily available in the encoded video.

The videos were encoded using the Berkeley MPEG encoder⁵. However, the results should hold for other MPEG encoders since the choice of encoder has little impact on compression relative to the impact on compression due to choice of quantization level and GOP pattern. The quantization values for I, P and B frames were set to 3 to yield a high picture quality in every frame. A commonly-used MPEG GOP pattern, 'IBBPBBPBBPBBPBB' and a typical full motion frame rate of 30 f/s were used. The encoded MPEG files are analyzed with the Berkeley `mpeg_stat` tool, obtaining the block types (skip, intra, zero motion or motion) and motion vectors.

3.1 Motion

For each video, each user visually divided the frame into 16 equal blocks and counted the motion characteristic of each block during the video. For each block, the user rated the amount of motion in that block along a 5-point scale: 0, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$ to 1, where 0 meant no motion and 1 meant lots of motion. As a simple example, if half of a block changed during half of the playout period, the blocks motion would have been rated as $\frac{1}{4}$. The scores of 16 blocks are averaged to get the motion score for the video for the user. The scores are averaged across users to get the overall motion score for the video.

Peker et al.'s [12] analysis of MPEG motion vectors suggests the variance of the motion vector magnitudes, similar to that provided in MPEG-7, is promising as a representative measure of visual motion. However, in our analysis the linear correlation with motion as perceived by the users is weak, only 0.51. Ma et al. [9] propose the perceived motion energy spectrum (PMES) as a measure of motion, obtained by a temporal energy filter and a global motion filter. In this method, a temporal energy filter is designed to disregard object motion in a scene, and a global motion filter is designed to shield object motions from camera motions. With PMES, the correlation with motion perceived by the users is only moderate, 0.70. Tripathi et al. [13] use the percentage of interpolated macroblocks in the B frames as a measure of motion. A high number of interpolated

macroblocks implies that a greater portion of the frame is similar to frames that are already existing in the stream (i.e. less motion) and a low number of interpolated macroblocks implies that there are a greater number of changes between frames (i.e. more motion). The correlation with motion perceived by the users is a moderate 0.63.

We propose a new measure of motion, the percentage of encoded macroblocks, i.e. the percentage of Forward/backward or Intra-coded Macroblocks (PFIM) in all frames. The intuition behind PFIM is that a video with visual changes from frame to frame will have these changes encoded (either by neighboring blocks or independently of other blocks), while video without visual changes can skip much of the encoding. Figure 3.1 shows the comparison of PFIM score and user score. Visually the correlation is significant, with the measured correlation being a strong 0.95.

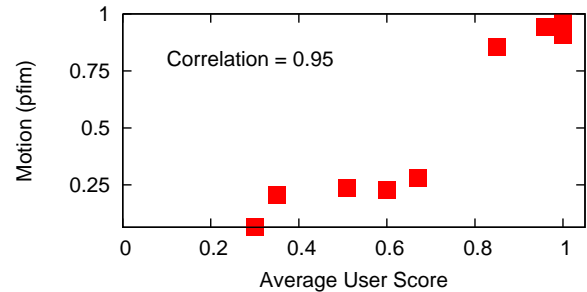


Figure 1: Percentage (fraction) of Forward/backward or Intra-coded Macroblocks (PFIM)

3.2 Scene Complexity

For each video, each user visually divided the frame into 16 equal blocks and determined the scene complexity of each block during the video. For each block, the user rated the complexity in that block along a 5-point scale: 0, 0, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$ to 1, where 0 meant a uniformly-colored block and 1 meant extremely complicated-looking block. As a simple example, if half of a block is of a single color and the other is varied, that blocks scene complexity would be rated as $\frac{1}{2}$. The scores of the 16 blocks are averaged to get the scene complexity score of the video clip.

We propose a new measure of scene complexity, the average of Intra-coded Block Size (IBS). If the scene is simple, there is not much information to be encoded, so the intra-coded block size will be small. If the scene is complicated, the intra-coded block size will be large to contain all the information. Figure 3.1 shows the comparison of IBS score and user score. Visually the correlation is modest, with a moderate measured correlation of 0.68. However, in the absence of a better metric, analysis of scene complexity for video games proceeds using IBS.

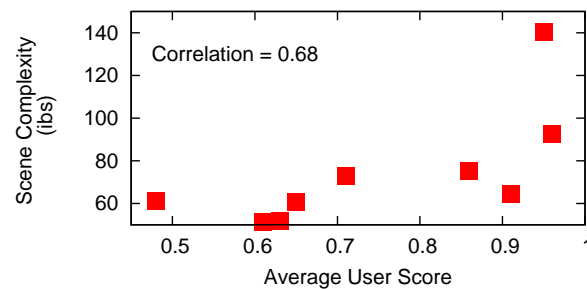


Figure 2: Average Intra-coded Block Size (IBS)

⁵<http://bmc.berkeley.edu/frame/research/mpeg/>

4. GAME PERSPECTIVES

There is evidence to suggest the system effects on playability are impacted by the *game perspective*. Game perspective is: (1) the placement of the camera with respect to the avatar in the game world, and (2) the visual change in object sizes relative to the position of the camera. Based on this definition, practically all computer games can be classified as one of the following types:

First person Linear Perspective. In the first-person linear perspective, the camera location is synonymous with the avatar's eyes and the game world objects appear smaller and closer together the farther they are from the camera location, giving the illusion of a 3-dimensional world in a 2-dimensional space. Examples of games with a first person linear perspective include *Doom*, *Half-Life*, *Need for Speed* and *Flight Simulator*. For the rest of this paper, the first person linear perspective is referred to as the first person perspective.

Third person Linear Perspective. In the third person linear perspective, the camera is placed at some point around the avatar, and the size and clustering of game objects varies with distance from the camera providing a 3-dimensional feel to the game. Examples of games with a third-person linear perspective include *World of Warcraft*, *Madden NFL*, and *Tomb Raider*.

Third person Isometric Perspective. In the third person isometric perspective, the camera is placed at some point around the avatar, but the size and clustering of game objects does *not* vary with distance from the camera, thus projecting a 2-dimensional world on a 2-dimensional space. Examples of games with a third person isometric perspective include *Diablo*, *Super Mario World*, and *Pacman*.

In addition, many games provide a player an overall view of the world, not just from the perspective of a single avatar. Such a viewpoint is in the third person, but has elements of both the isometric and linear perspective depending upon the panning and zooming actions of the player. Thus, we define a fourth perspective type:

Omnipresent Perspective. In the omnipresent perspective, players view and simultaneously influence the entire set of resources under their control. The size and clustering does not vary unless players zoom in on a region of interest.

5. METHODOLOGY

The following methodology was used to analyze the characteristics of game videos and provide preliminary performance evaluation for thin clients: 1. Select games (Section 5.1); 2. Record traces (Section 5.2); 3. Select videos (Section 5.3); 4. Analyze data (Section 6); and 5. Evaluate thin clients (Section 7).

5.1 Select Games

Based on the perspectives presented in Section 4, 29 games were selected from a variety of perspectives, developers, and generations. The games and perspectives are shown in Table 1.⁶

5.2 Capture Game Videos

The game videos were captured using FRAPS⁷, a video capture and benchmarking tool that records frames from games using DirectX or OpenGL graphic technologies. FRAPS v2.7.2, build 5534 was used, capturing full-sized video at 30 f/s, with no cursor or

⁶Details on version numbers for each game can be found at: <http://www.cs.wpi.edu/claypool/papers/game-motion/>.

⁷FRAPS, by Beepa, is online at <http://www.fraps.com/>

Table 1: Selected Games

Perspective	Game
First	Battlefield 1942, Battlefield 2, Battlefield Vietnam, Doom 3, Medal of Honor Allied Assault, Quake III Arena, Star Wars Battlefront
Third (Lin)	Fahrenheit, Guild Wars, Harry Potter Chamber of Secrets, The Incredibles, The Wonderful End of the World
Third (Iso)	Diablo II, Evil, Galactic Mail, Koalabr8 Lazarus, Pyramid Panic, Rainbow Reef, Wingman Sam
Omnipresent	Age of Empires 3, Age of Mythology, Battle for Middle Earth 2, Command and Conquer 3, Command and Conquer Generals, Company of Heroes, Star Wars Galactic Battlegrounds, Stronghold 2, Warcraft III

sync. FRAPS was manually started at the beginning of a game sequence and manually stopped after at least 30 seconds of game play. Captures longer than 30 seconds were truncated to exactly 30 seconds during analysis, except where otherwise noted. FRAPS produces videos in the audio video interleave (.avi) format, whereupon the Berkeley MPEG tools⁸ are used for subsequent content analysis.

5.2.1 Resolution Justification

All captures were made on a Windows XP (with service pack 2) PC with an Intel Pentium 4, 4.0 GHz CPU with 512 MB RAM and an nVidia Geforce 6800GT 256 MB VRAM graphics card. In practice, games are often played at a range of screen resolutions so, in theory, for benchmarking, capturing video at the highest resolution supported by the hardware would allow subsequent analysis at a range of lower resolutions. However, frame capture by FRAPS incurs some overhead as saved frames are written to disk, making it difficult for the frame capture to keep up with the game engine at high resolutions. In order to determine the highest resolution that is still low enough to allow the frame capture to keep up with the game engine, the built-in demo for Doom 3 was used to benchmark the video card. Doom 3 is produced by id Software⁹, and provides unified lighting and shadowing, and complex animations with real-time, fully dynamic per-pixel lighting and stencil shadowing - all parameters that can stress a graphics system. The Doom 3 demo replays a game wherein a player battles a dozen or so monsters with various weapons over the course of about a minute.¹⁰ The first run of each demo was discarded to mitigate cold-start effects. For each screen resolution available by the Doom 3 game engine, the FRAPS benchmarking utility was used to record the minimum, average, and maximum frame rates. The experiments were then repeated while capturing the game videos. Figure 3 depicts the average frame rates

From the data, the PC used for capture can support high frames rates (30 f/s) at all tested resolutions when not capturing the game video to disk. However, when capturing game video, there is an appreciable drop-off in frame rate when resolutions are above 800x600 pixels. Thus, 800x600 pixels is chosen for all subsequent game video captures.

5.2.2 Graphics Justification

Most games have a variety of graphics options to complement

⁸<http://bmc.berkeley.edu/frame/research/mpeg/>

⁹<http://www.idsoftware.com>

¹⁰From Doom 3, the game console is opened by pressing ctrl+alt+~, and the demo run by typing "playdemo demo1".

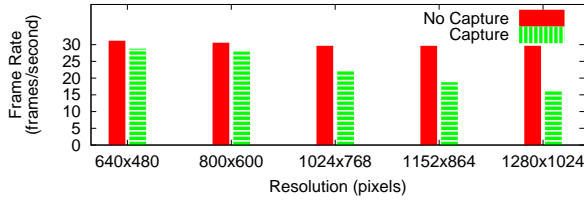


Figure 3: Frame Rate with and without Capture

the screen resolution chosen. For example, Doom 3 has settings for shadows, specular lighting, bump maps, sync and anti-aliasing. Like resolution, these settings can have a negative impact on game video capture. In order to ensure graphics settings did not reduce frame rates during capture, with a resolution of 800x600, all advanced graphics were enabled and the FRAPS benchmarking utility used to record the minimum, average, and maximum frame rates while capturing the Doom 3 game video. With these settings, the PC was still able to maintain an average of 28.5 f/s (minimum 15 f/s, maximum 33 f/s). However, since not all game engines support the same graphics options Doom 3 has, graphics options are set to “Medium” for all subsequent game video captures.

5.2.3 Length Justification

Accurate representation of a game suggests a long video length, while reducing the resources (time and space) required for experimentation suggests a short video length. Three games were selected, one from each of the first, third and omnipresent perspectives (Battlefield 2, Guild Wars, and Battle for Middle Earth, respectively) in order to ascertain an appropriate video length for testing. A five minute trace for each game was captured, and the motion (PFIM) for one second averaged for slices from 1 up to 60 seconds. A 95% confidence interval was computed for each slice length, depicted in Figure 5.2.3.

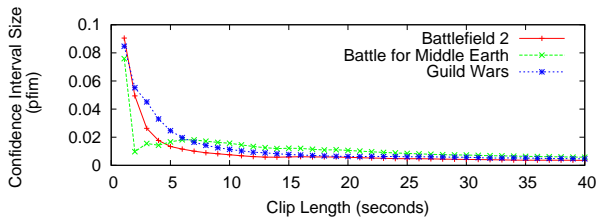


Figure 4: Confidence Interval Size (PFIM) for Three Games

The confidence interval size decreases as the clip length increases since the number of samples increases. The “knee” in each of the curves flattens out around 15-20 seconds. Thus, 30 seconds is selected for the length for each clip since it provides a small (less than 0.005 pfim) confidence interval in each case.

In addition, uncontrolled system events may cause variance in the video captured, thus impacting measures of motion and scene complexity even within the same game. In order to ascertain the variance caused by the system, videos were captured for the Doom 3 demo run four separate times, then analyzed for motion and scene complexity. Table 2 depicts the results. The variation across runs is quite small, considerably less than the variations within the game itself. Thus, it is assumed one 30-second trace is long enough to effectively dwarf variance caused by the system.

5.3 Select Videos

Video clips were selected in order to compare game videos with video that is more regularly used for streaming. The video clips

Table 2: Variation on Motion and Scene Complexity

Run	PFIM	IBS
1	0.877	468.5
2	0.876	466.0
3	0.878	464.1
4	0.879	461.7

were chosen from among those commonly used by the multimedia community and represent a range of motion and scene complexities. These videos are each 10 seconds long, recorded at 30 frames per second in CIF format (352x288 pixels). The videos, with a brief description, are listed in Table 3.

Table 3: Selected Videos

Video	Description
Coastguard	Panning of a moving coastguard ship
Container	A container ship sailing slowly
Foreman	A close up of a talking head
Hall	An office hallway with some people
Mobile	Panning of moving toys
News	Two news reporters talking
Paris	Two people talking with gestures
Silent	A person demonstrating sign language
Vectra	Panning of a moving car

6. ANALYSIS

The motion (PFIM) and scene complexity (IBS) for each game is computed and provided in tabular form in Table 4 and graphed in a scatter-plot in Figure 5. For comparison, the representative video clips are depicted as well, shown with small, filled squares.

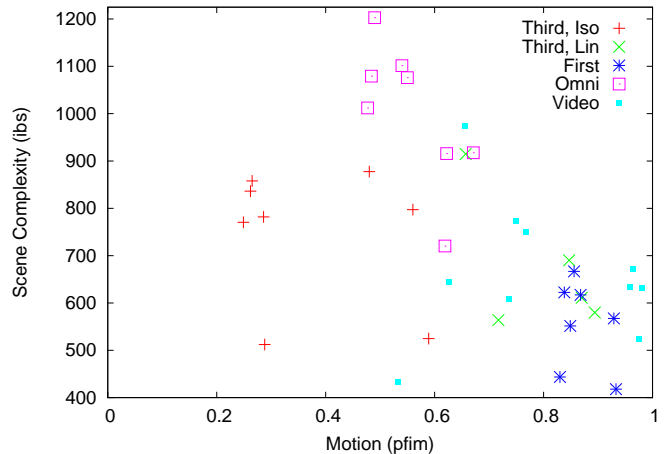


Figure 5: Scatter Plot of Scene Complexity (IBS) and Motion (PFIM) for All Games (videos also shown for comparison)

6.1 Motion

The amount of visual motion varies considerably across all games, from about 0.2 to about 0.95, a considerably broader range than for traditional videos which vary only from about 0.7 to just under 1.

First person games have considerably higher motion than do omnipresent and third person linear games, likely because of the frequent panning effect of the background as the player moves through

Table 4: Scene Complexity (IBS) and Motion (PFIM) for All Games (videos shown at the bottom for comparison)

Perspective	Game	PFIM	IBS
First	Battlefield 1942	0.838	622.3
	Battlefield 2	0.856	666.7
	Battlefield Vietnam	0.868	617.0
	Doom 3	0.830	443.6
	Medal of Honor Allied Assault	0.849	551.5
	Quake III Arena	0.933	417.9
	Star Wars Battlefront	0.929	567.4
Third (Lin)	Fahrenheit	0.717	563.9
	Guild Wars	0.657	914.6
	Harry Potter Chamber Secrets	0.870	610.9
	The Incredibles	0.894	579.8
	Wonderful End of the World	0.847	690.2
Third (Iso)	Diablo II	0.560	797.2
	Evil	0.286	781.8
	Galactic Mail	0.288	512.1
	Koalabr8	0.249	770.4
	Lazarus	0.265	857.9
	Pyramid Panic	0.480	877.6
	Rainbow Reef	0.262	836.3
	Wingman Sam	0.589	524.8
Omnipresent	Age of Empires 3	0.490	1202.7
	Age of Mythology	0.540	1101.4
	Battle for Middle Earth 2	0.610	917.4
	Command and Conquer 3	0.550	1076.1
	Command and Conquer Gen.	0.622	915.7
	Company of Heroes	0.619	720.4
	Star Wars Battlegrounds	0.454	1619.1
	Stronghold 2	0.477	1012.0
	Warcraft III	0.484	1079.3
Videos	Coastguard	0.981	632.2
	Container	0.767	750.1
	Foreman	0.958	634.3
	Hall	0.736	607.7
	Mobile	0.965	672.4
	News	0.626	645.2
	Paris	0.655	974.1
	Silent	0.750	772.7
	Vectra	0.976	524.3

the world. Third person isometric games have the lowest amount of motion, owing to the relatively static nature of the visual game world as play progresses. However, a few third person linear games have medium motion, notably those that are side scrolling games where background scenery in the world pans behind the character to provide an illusion of movement. Omnipresent games consistently have a medium level of motion, probably because of the top-down view of the world with some panning and animated units as the world is managed. Third person linear games generally have high levels of motion, slightly higher than the motion in omnipresent games but slightly lower than the motion in first person games. Generally, traditional videos have about the same amount of motion as do third person linear games. Motion in traditional videos is typically more than for omnipresent and third person isometric games but less than for first person games. However, traditional videos with the most motion have more motion than any other game.

6.2 Scene Complexity

Scene complexity varies considerably across all games, from about 400 to about 1200, a range mostly matched by traditional videos which vary from about 400 to just under 1000.

First person games have the least complex scenes. This may be because responsiveness is valued for game play over visual de-

tail [4]. Omnipresent games have the most complex scenes. This is likely because omnipresent games typically provide the player with lots of details on the state of objects in the world. Third person isometric games have mostly medium scene complexity, but a few have simple scenes, while third person linear games have mostly simple scenes with a few games tending towards medium scene complexity. The scene complexity for traditional videos varies a lot, from quite low to medium-high, but the most complex traditional videos still have lower scene complexity than the most complex games.

6.3 Summary

As a summary, the conceptual characterization of motion and scene complexity for games is abstracted into a visual representation, shown in Figure 6 and in tabular form, shown in Table 5.

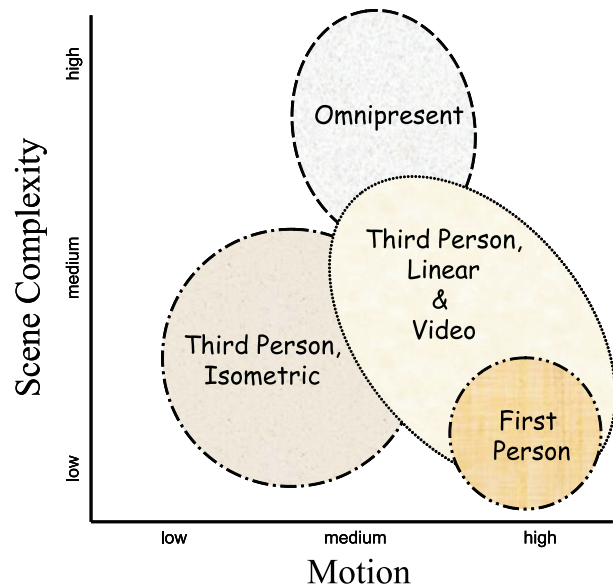


Figure 6: Visual Summary of Scene Complexity and Motion

Table 5: Tabular Summary of Scene Complexity and Motion

Perspective	Motion	Scene Complexity
First	High	Low
Third, Linear	Low-Medium	Low-Medium
Third, Isometric	Medium-High	Low-Medium
Omnipresent	Medium	Medium-High
Video	Medium-High	Low-Medium

6.4 Variation within a Game

Game scene content can vary fairly widely during a scene. While 30 seconds is used for representative analysis (see Section 5.2.3), three games were selected, one from each of the first, third and omnipresent perspectives (Battlefield 2 (BF2), Guild Wars (Guild), and Battle for Middle Earth (BFME), respectively) in order to get a better understanding of 30-second motion variation within a longer game. Figure 7 depicts the results. BF2 and Guild exhibit considerable variation, ranging from about 0.65 to 0.82 for BF2 and from about 0.60 to 0.79 for Guild. BFME exhibits the least variation, probably attributed to the more consistent nature of the view and camera angle afforded by an omnipresent perspective.

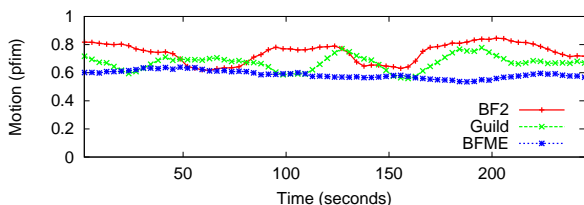


Figure 7: Motion (PFIM) versus Time for Three Games

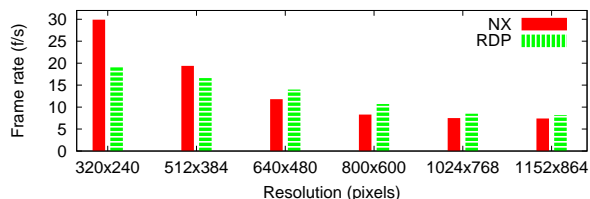


Figure 8: Frame Rate for a First Person Game

7. THIN CLIENT EVALUATION

In this section, the intent is to provide a preliminary assessment of existing thin client technologies to support streaming video games. The goal is not to provide a rigorous, head-to-head comparison of all possible thin client options but rather to present a brief look at performance issues that may arise when streaming games.

Many computer games cannot actually be played using a thin client since the graphics drivers need direct access to the memory on the video card. Modifications to thin client architectures to bypass these graphics issues, such as proposed in [16], are effective but not yet widely implemented. Instead, for our evaluation, game video is played using the VideoLAN VLC¹¹ media player, v0.8.6f. The server is listed in Section 5.2.1 and the thin client runs on a Sony Vaio laptop, running Windows XP (sp2) with an Intel Pentium M, 2.26 GHz CPU with 2 GB RAM and an nVidia GeForce Go 6400 64 MB graphics card. Both server and client are connected by 802.11g wireless running in a position of excellent signal strength. Wireshark¹² v0.99.4 is used to capture traffic. For performance, VLC reports frame playout statistics and Wireshark enables processing of network traffic.

Our experiments focused on two popular thin clients, Microsoft's Terminal Services (which uses the Remote Desktop Protocol, or RDP) and NoMachine's NX Client for Windows, v3.1.0-3, which uses X11. NX compresses the 24-bit color X11 data and makes use of caching to reduce the number of round-trip times from the client to the server. RDP does low level graphics with a server-push, lazy update, with compression and 8-bit further to reduce the bitrate requirements. Both NX and RDP run over TCP (NX runs over SSH). Both NX and RDP have options for tuning performance for networks ranging from a dialup modem to a Local Area Network (LAN), but all our experiments use just the LAN setting.

In order to evaluate performance at different game resolutions, the built-in Doom 3 demo was used with different display settings: 320x240, 512x384, 640x480, 800x600, 1024x768, and 1152x864 pixels.

Figure 8 depicts a comparison of the displayed frame rates for NX and RDP at different resolutions. For both NX and RDP there is a consistent drop in frame rate as the resolution increases. NX performs slightly better than RDP at low resolutions and slightly worse at higher resolutions. Both thin clients are able to maintain acceptable frame rates (above 15 f/s, based on an earlier user study [3]) for only the lowest resolutions.

Figure 9 depicts a comparison of the downstream (server to client) bitrates for NX and RDP at different resolutions. When streaming video games, both NX and RDP thin client technologies require bitrate requirements above typical residential broadband connections. In nearly all cases, RDP has substantially higher bitrates than does NX. While there is no clear visual relationship between game resolution and bitrate, the lowest resolution (320x240) has the lowest

bitrate for both NX and RDP. Packet sizes also do not show obvious correlation with resolution, with average packet sizes of 1163 bytes and 1095 bytes for NX and RDP, respectively.

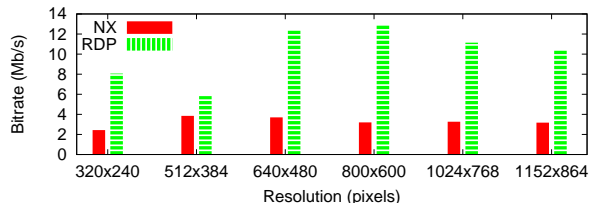


Figure 9: Bitrate for a First Person Game

In order to evaluate performance for different game perspectives, one game at 800x600 pixel resolution was selected from 3 different genres: third person isometric (Koalabr8), omnipresent (BFME), and first person (BF2). Figure 10 (left) depicts a comparison of the displayed frame rates for NX and RDP for the different games and Figure 10 (right) depicts a comparison of the downstream bitrates. There is some correlation between motion and frame rate, with higher motion (first person) generally resulting in a lower frame rate. There is less correlation with scene complexity as the most complex game, the omnipresent, has a frame rate similar to the much more simple third person isometric. There is little correlation for either thin client bitrate and motion or scene complexity.

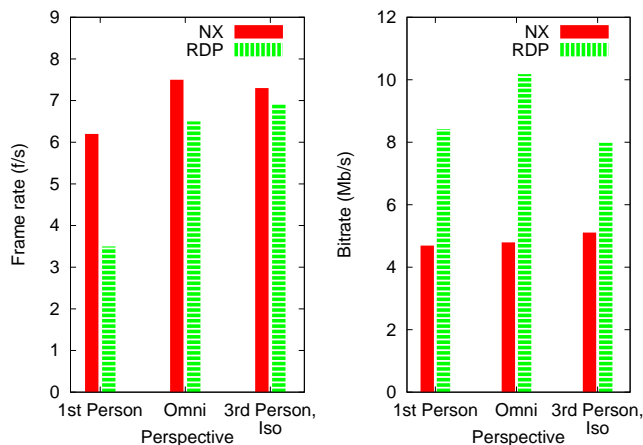


Figure 10: Frame Rate (left) and Bitrate (right) for Different Games at 800x600 Pixel Resolution

8. CONCLUSIONS

The growth in capacity and connectivity of networks presents the opportunity for thin clients, traditionally used for applications with relatively static displays and limited interaction, to be used

¹¹<http://www.videolan.org/>

¹²<http://www.wireshark.org/>

for streaming video games. Thin clients for games can be used for: complex game scenes shown at underpowered clients; for augmented reality games using wearable computers; and for easing maintenance costs of large scale, multiplayer games. However, in order to make significant advances in streaming video games, the visual characteristics (motion and scene complexity) of games must first be better understood and the performance limitations of existing thin client technologies quantified.

This paper presents a first step at addressing this research area with 4 main contributions. 1) The set of 29 game videos used in this study is a substantial research contribution. The intent is for these videos to serve as a public benchmark for evaluating performance in subsequent streaming game research. Online¹³ are the compressed MPEG (.mpg) files, compression information, and analysis data. In addition, scripts used to compute motion (PFIM) and scene complexity (IBS) are provided. The audio video interleave (.avi) files are also available by request,¹⁴ but are not kept online because of their size (over 30 Gbytes total). 2) Novel metrics for motion and scene complexity, namely the percentage of forward/backward or intra-coded macroblocks (PFIM) and the average of intra-coded block size (IBS), are presented and used to analyze the game videos. 3) Preliminary evaluation of two popular thin client technologies is provided, highlighting performance in terms of bitrate and frame rate for representative games and a range of display resolutions. And 4), overall, this work can be used to answer the questions posed in the Introduction (Section 1):

1. Typical video encoding characteristics can be used in an objective measure of motion and scene complexity that correlates with user perception. Specifically, the percentage of forward/backward or intra-coded macroblocks (PFIM) and the average of intra-coded block size (IBS) provide reasonable representations of motion and scene complexity, respectively, as perceived by users.
2. Motion and scene complexity characteristics vary considerably across computer games. The perspective of the game camera, whether first or third person, impacts both the amount of motion and the scene complexity. First person games all have higher motion than omnipresent and third person linear games, while third person isometric games have the lowest amount of motion.
3. Motion and scene complexity characteristics are different for games as compared to video. Games vary more than typical videos in both amount of motion and in scene complexity, with omnipresent games tending to have more complex scenes than most videos and third person linear games tending to have less motion.
4. Streaming video games from a powerful server to a thin client is viable with today's thin client technology only for games with low motion using very low display resolution. Bitrate requirements are higher than typical residential broadband capacities, but can be met on a typical LAN. Even when there is sufficient capacity, however, high display resolution and/or high motion cause frame rates to drop to unacceptable levels.

The measures of motion in this paper can be leveraged into a working thin client system that supports games with the knowledge about typical game characteristics being useful in deciding upon temporal or quality scaling for streaming video games.

Previous research has shown that latency can significantly degrade performance for games played over a WAN. Thus, another potential area for future work is the study of latency and thin clients.

Acknowledgements

I would like to thank Huahui Wu for his help on the user study and discussions on measures of motion.

9. REFERENCES

- [1] M. Ayromlou, M. Zillich, W. Ponweiser, and M. Vincze. Measuring Scene Complexity to Adapt Feature Selection of Model-Based Object Tracking. *Springer Computer Vision Systems*, pages 448–459, 2003.
- [2] M. Billinghamurst, S. Bee, J. Bowskillb, and H. Kato. Asymmetries in Collaborative Wearable Interfaces. In *Intl. Symposium on Wearable Computers (ISWC)*, 1999.
- [3] K. Claypool and M. Claypool. On Frame Rate and Player Performance in First Person Shooter Games. *ACM/Springer Multimedia Systems Journal (MMSJ)*, 2007.
- [4] M. Claypool, K. Claypool, and F. Damaa. The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games. In *ACM MMCN*, San Jose, CA, USA, 2006.
- [5] J. Kim, R. A. Baratto, and J. Nieh. pTHINC: A Thin-Client Architecture for Mobile Wireless Web. In *World Wide Web Conference (WWW)*, Edinburgh, Scotland, May 2006.
- [6] A. Lai and J. Nieh. On the Performance of Wide-Area Thin-Client Computing. *ACM Transactions on Computer Systems*, 24(2):Pages 175 – 209, May 2006.
- [7] F. Lamberti and A. Sanna. A Streaming-Based Solution for Remote Visualization of 3D Graphics on Mobile Devices. *IEEE Transactions on Visualization and Computer Graphics*, Mar/Apr 2007.
- [8] Z. Lei and P. Bhattacharya. Scene Complexity Analysis using Random Curves. In *Computer Graphics and Imaging*, 2008.
- [9] Y. Ma and H. Zhang. A New Perceived Motion based Shot Content Representation. In *IEEE Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001.
- [10] K. Ocheltree, S. Millman, D. Hobbs, M. McDonnell, J. Nieh, and R. Baratto. Net2Display – A Proposed VESA Standard for Remoting Displays and I/O Devices over Networks. In *ADEAC*, Atlanta, GA, USA, Oct. 2006.
- [11] K. Packard and J. Gettys. X Window System Network Performance. In *USENIX Annual Technical Conference*, San Antonio, TX, USA, 2003.
- [12] K. Peker and A. Divakaran. Framework for Measurement of the Intensity of Motion Activity of Video Segments. *Elsevier JVCIR*, 2004.
- [13] A. Tripathi and M. Claypool. Improving Multimedia Streaming with Content-Aware Video Scaling. In *Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Durham, NC, USA, Mar. 2002.
- [14] D. Tynan. Think Thin. *InfoWorld*, July 2005.
- [15] Wikipedia. Remote Play. Online. http://en.wikipedia.org/wiki/Remote_Play.
- [16] D. D. Winter, P. Simoens, and et al. A Hybrid Thin-client Protocol for Multimedia Streaming and Interactive Gaming Applications. In *NOSSDAV*, Newport, RI, USA, June 2006.
- [17] H. Wu, M. Claypool, and R. Kinicki. On Combining Temporal Scaling and Quality Scaling for Streaming MPEG. In *NOSSDAV*, Newport, RI, USA, June 2006.

¹³Links for downloading are available at:
<http://www.cs.wpi.edu/~claypool/papers/game-motion/>

¹⁴Send email to claypool@cs.wpi.edu.