An Experiment to Characterize Videos Stored On The Web

Soam Acharya
Department of Electrical Engineering
Cornell University
Ithaca, NY 14853
soam@ee.cornell.edu

Brian Smith
Department of Computer Science
Cornell University
Ithaca, NY 14853
bsmith@cs.cornell.edu

Abstract

The design of file systems is strongly influenced by measuring the use of existing file systems, such file size distribution and patterns of access. We believe that a similar characterization of video stored on the Internet will help network engineers, codec designers, and other multimedia researchers. We therefore executed an experiment to measure how video data is used on the Web today. In this experiment, we downloaded and analyzed over 57000 AVI, QuickTime and MPEG files stored on the Web -- approximately 100 Gigabytes of data. Among our more interesting discoveries, we found that the most common video technology in use today is QuickTime, and that the image resolution and frame rate of video files that include audio are much more uniform than video-only files. The majority of all audio/video files have dimensions of CIF or QCIF (or very similar) at 10, 12, 15, or 30 fps, whereas the dimensions and frame rates of video-only files are more uniformly distributed. We also experimentally verified the conjecture that current Internet bandwidth is at least an order of magnitude too slow to support streaming playback of video. We present these results and other statistical information characterizing video on the web in this paper.

1. Introduction

In 1985, researchers at the University of California at Berkeley published a study of the UNIX 4.2 BSD file system [1]. This analysis provided a number of insights regarding file sizes, lifetimes and access and was highly influential in the design of several file systems. In 1991, Berkeley researchers released a follow-up study on Sprite [2], which verified the assertions made in [1] and made further measurements. This study was also influential in the design of subsequent distributed file systems. These two papers, and the subsequent activities, they inspired provide a good example of the basic work necessary to develop an effective system: an understanding of the data used in the system and how that data is accessed.

The use of video on the Web has reached the point where a similar study can be meaningfully conducted. We therefore wrote a Web crawler to download and analyze every video we could find on the Web. Our goal was to answer the following questions:

- What are the basic properties (size, frame rate, picture dimensions, duration, and average bitrate) of video files?
- How are these characteristics changing?
- How do different compression technologies compare with each other in practice?
- How does network bandwidth affect the waiting times for movie file download?

Our answers to these questions are based on 49,000 video data files downloaded from about 9300 WWW servers in April and May 1997. Our findings included:

1. Movie sizes range from hundreds of kilobytes to several megabytes: 1.2MByte is a good rule of thumb.
2. Most movies are brief: 90% last 45 seconds or less.
3. Users adhere to “standard” small or medium picture dimensions (such as 160x120 or 320x240) when creating videos with audio content, but not when creating videos without audio.
4. The number of movies coming on-line is increasing exponentially.
5. MPEG [7] files compress better than QuickTime [6] or AVI [5] due to their lower bits/pixel values. MPEG files also have smaller playback times and higher frame rates.

6. QuickTime and AVI files share many similar properties. They have analogous distributions for frame rates, duration, size and waiting time. Bits/pixel comparisons show QuickTime to compress slightly better.
7. 28.8K, 56K and 128K bandwidths are useless for real-time display of video data. For example, 56Kbits/sec allows about 1% of all movie files to be downloaded. About half the movies can be displayed with 700 Kbits/sec of bandwidth, 80% with 1.5 Mbits/sec and 90% with 2 Mbits/sec.

The rest of this paper is organized as follows. Section 2 presents our method for locating, collecting, and processing the video data. Section 3 presents the results of eleven analyses we performed on our collected data set, elaborating the results outlined above. We outline related work in section 4 and summarize our conclusions in section 5.

2. Video Data Collection Process

For our study to be representative of conditions on the Web, we had to locate and analyze a large number of video files. We divided this task, the video data collection process, into three steps. The first step, the hunting phase, was to obtain a list of links to *video documents*. A video document is an HTML document that contains at least one link to a video clip. The second step, the gathering phase, consisted of extracting the video links from each video document, fetching the specified clip, analyzing it, and recording the results. We found a small amount of the data was suspicious, so the final step, the sifting phase, eliminated this suspicious data. The next three subsections describe these steps in detail.

2.1 Step 1: The Hunting Phase

To obtain a list of video documents, we wrote a Tcl script [4] to coerce the Alta Vista search engine [4] to return a list of potential video documents. A potential video document is one that contains a link to an MPEG, QuickTime, or AVI file - we used file extensions of URLs to distinguish between video and non-video links. We limited our search to AVI, MPEG and QuickTime files since these are the most established video technologies. Alta Vista is capable of ordering query results by the date of last modification and we employed this facility to categorize the retrieved video document links on a month by month basis, from January 1995 to March 1997. The process yielded about 44,000 links, of which 22,600 turned out to be valid links.

2.2 Step 2: The Gathering Phase

Armed with the list of potential video documents, we wrote a system to download the video documents, and contained video files. Our video link processing system, shown in figure 1, consists of a link distributor and gatherer (*LDG*) process and a set of link processor (*LP*) process. The LDG is responsible for assigning video documents to LPs and collecting and storing the summary statistical data calculated by the LPs. Upon obtaining a video document URL, the LP fetches the document, parses it to extract any links to movie clips, downloads the clips, runs a video analysis program, and sends summary statistics to the LDG. These statistics include the basic movie properties (frame rate, clip size in bytes, etc) as well as properties of the video document (modification time, size in bytes).

For example, suppose the video document http://www.eg.com/movie.html contains a link to http://www.hoho.com/my.mov. Figure 1 illustrates the steps the video link processing system would take in processing this video document:

1. LP_1 requests a new video document link from the LDG. It receives the URL http://www.eg.com/movie.html
2. LP_1 contacts www.eg.com and fetches movie.html
3. LP_1 parses the contents of movie.html and extracts the link http://www.hoho.com/my.mov.
4. LP_1 contacts www.hoho.com and downloads my.mov.
5. LP_1 spawns a program to analyze my.mov and collects the results.
6. LP_1 contacts the LDG and reports the statistics on my.mov and movie.html.

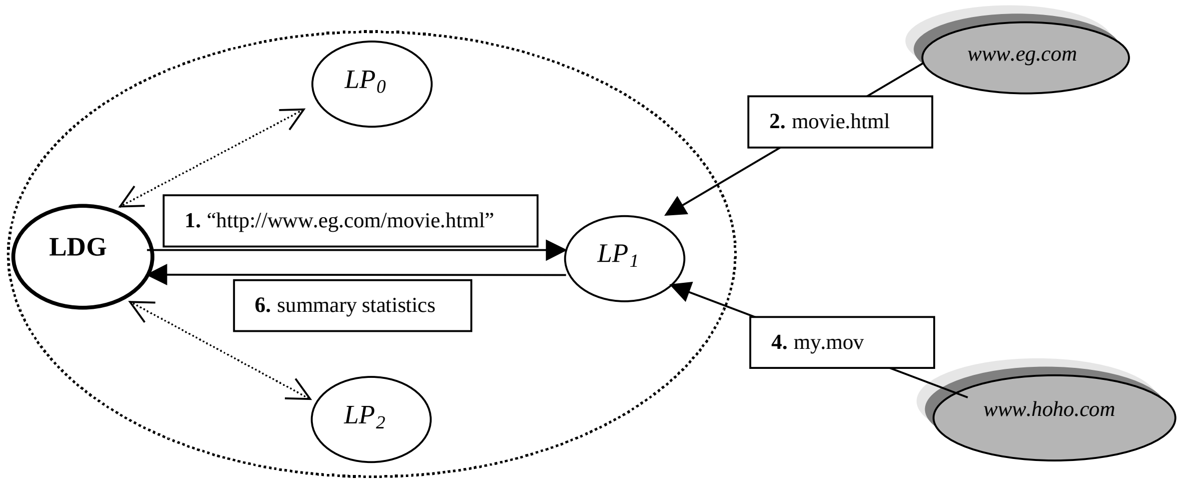


Figure 1: Video Link Processing Architecture

Each LP and LDG process runs on a separate UNIX workstation. In our experiment, eight machines ran the LP processes and one machine ran the LDG. It took about six weeks to download and analyze all 44,000 potential video documents. Of these, about 22,600 were valid links. The rest either did not exist or did not contain links to video data. Not surprisingly, link processing was faster at night due to lower Internet traffic. We wrote the core portions of the LP and LDG in Tcl, employing Tcl-DP [8] for communication between the LPs and LDG. We used *mpegstat* [9] to analyze the MPEG files, and an instrumented version of *xanim* [10] to analyze the QuickTime and AVI files.

About 10% of all the QuickTime, 8% of the AVI and 5% of the MPEG titles were not analyzed for one of more of the following reasons:

- There were QuickTime files that had not been "flattened"[1]
- xanim did not have the right QuickTime or AVI codec for the file
- The file was audio only
- The file had been truncated. A file was truncated if its downloaded size was not within 95% of the value claimed by the accompanying HTTP header.

2.3 Step 3: The Sifting Phase

In all, we downloaded and processed about 100GB worth of HTML and video files, accumulating 25MB of raw statistics. From this initial list of about 57000 titles, we excluded about 8,000 suspect videos using the following guidelines:

- *4 <= fps <= 40*.
  Files with frame rates less the four frames per second (fps) or greater than forty fps were excluded. We found many files to have a frame rate of 0.1 fps (some were zero), and others with a frame rate as high as 1000 fps. To avoid skewing the derived characteristics, such as movie duration (frame divided by fps), we eliminated these titles. This criterion eliminated about 5000 entries.
- *duration >= 0.5 seconds*
  Clips less than one-half second in duration were eliminated. About 1000 links were eliminated this way.
- *0.6 <= AR <= 1.6667*
  The aspect ratio (AR = width/height) was constrained to be in this range to conform to acceptable norms for video. Spot-checking revealed that videos with aspect ratios outside this range were largely collections of images, rather than true motion video. Nearly 1000 links deviated from this guideline.
- Bitrate < 10 Mbits/sec
  We define the bitrate of a movie clip as:

  Bitrate = movie size (bits)/movie duration (seconds).

[1] "Flattening" is a process that combines the resource and data forks of a QuickTime file thus making it portable.

We constrained the bitrate to be less than or equal to 10Mbits/sec. A bigger bitrate was unrealistic for real-time data transfer over the network. Almost 1000 files exceeded this threshold[2].

- Duplicate investigation of the same URL.
  We were careful to avoid downloading and analyzing the same URL more than once: before analyzing a movie link, we checked it against other already processed URLs. However, we did not account for instances such as DNS aliasing where one machine can be referred to via multiple names. For example http://cnn.com points to the same location as http://www.cnn.com. Hence, in this section, we compared IP addresses, file names and sizes to eliminate about 1500 duplicate titles.

After completing this process, 47,500 titles remained. This working data set consisted of 53% QuickTime files, 30% MPEG files, and 17% AVI files.

3. Results and Analysis

We analyzed the collected data using Microsoft Excel and Tcl scripts. We calculated four types of properties from this raw data, which are detailed in the following subsections:

- **Directly measurable quantities**, such as date of creation (section 3.1), frame rate (3.2), and movie size (3.3),
- **Derived quantities**, such as how average movie size is changing (section 3.4), movie duration (3.5), aspect ratio (3.6),
- **Codec properties**, which shows how the AVI (section 3.7), QuickTime (3.8), and MPEG (3.9) codecs are used to encode video for the Web, and
- **Network properties**, where we calculated the bandwidth required for streaming (3.10).
- **Replication**, where we measured how many movies are replicated on the Web (3.11)

3.1 Movie Date Distribution

A video file has two associated dates: its on-line date and document date. The on-line date of a video is the date it was placed on-line. The document date is the last modified time of the associated video document as reported by the Web server. Figure 2 plots the number of movies placed on the Web in a given time, using on-line dates. The growth is increasing until May 1996, after which the growth levels-off and declines. This behavior raises two questions:

1) Why are there movies dated April and May 1997 when the cutoff date of our initial survey was March 1997?
2) Why is there a decline of movies coming on-line from December 1996 – May 1997?

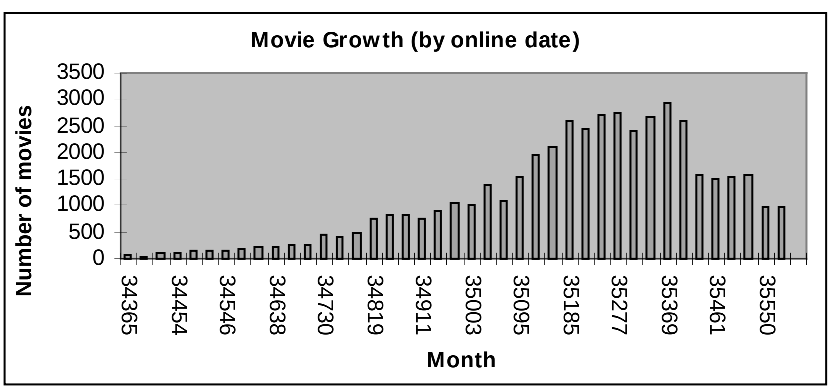


Figure 2: Movies added to the Web

[2] Yes, 1000 movie files have a bitrate of more than 10 Mbits/sec. The largest bitrate was 12.5 Gbits/sec

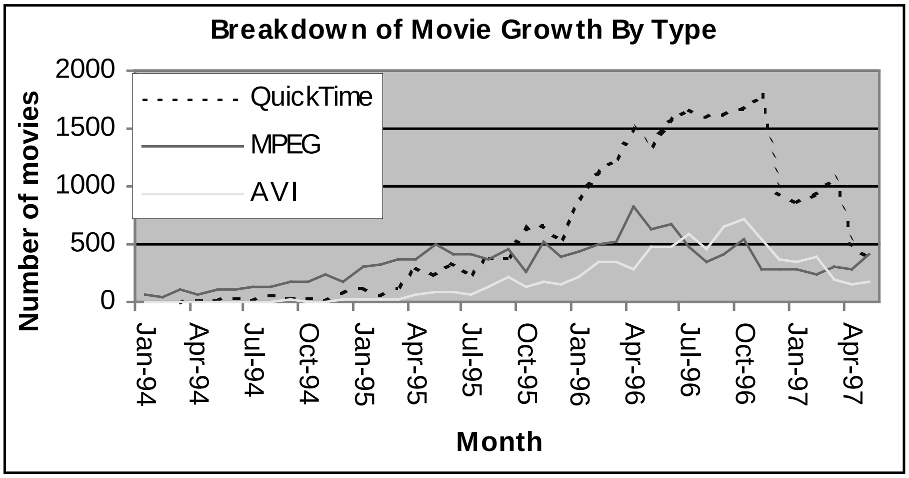


Figure 3: Number of movies added to the Web, by format

The answer to the first question is simply that the video documents were modified after we retrieved the document URL from Alta Vista. For example, many of the video documents are indices with dozens of movies. Suppose Alta Vista indexed such a document in March 1997, and that the author added a new movie in early May. That movie's on-line date would be correctly reported as May 1997.

The answer to the second question is more complex. Recall that we obtained our list of video documents from Alta-Vista. If we assume that Alta-Vista takes, say, six months to index the Web, then documents older than 6 months are certain to have been indexed. Documents that are more recent have a decreasing probability of being indexed. However, if Alta-Vista has not indexed the (HTML) video document, our search strategy would not find this video. For instance, if a video is placed on the Web in February 1997, but the associated video document is not indexed by Alta-Vista when we collect the list of potential documents, we will not find it. Thus, we are under-reporting the number of video files on the Web for dates close to March 1997.

Figure 3 decomposes figure 2 into the three movie types. QuickTime is clearly dominant format today, although MPEG led until mid 1995. AVI, initially the least used of the three formats, is currently comparable to MPEG in

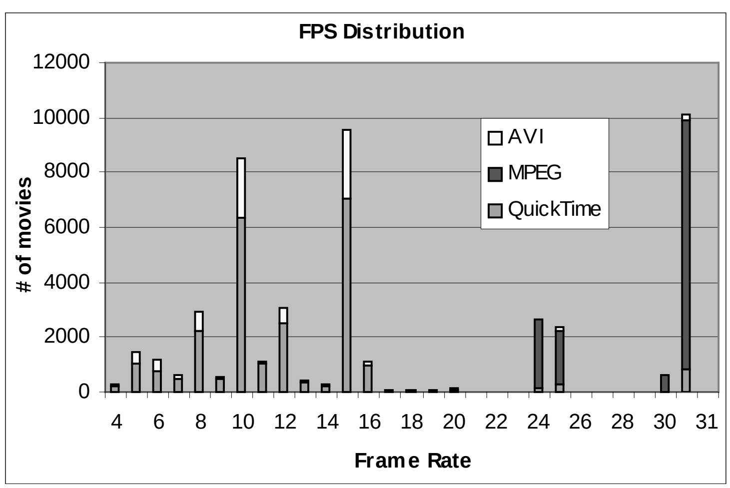


Figure 4: Frame rates of video files

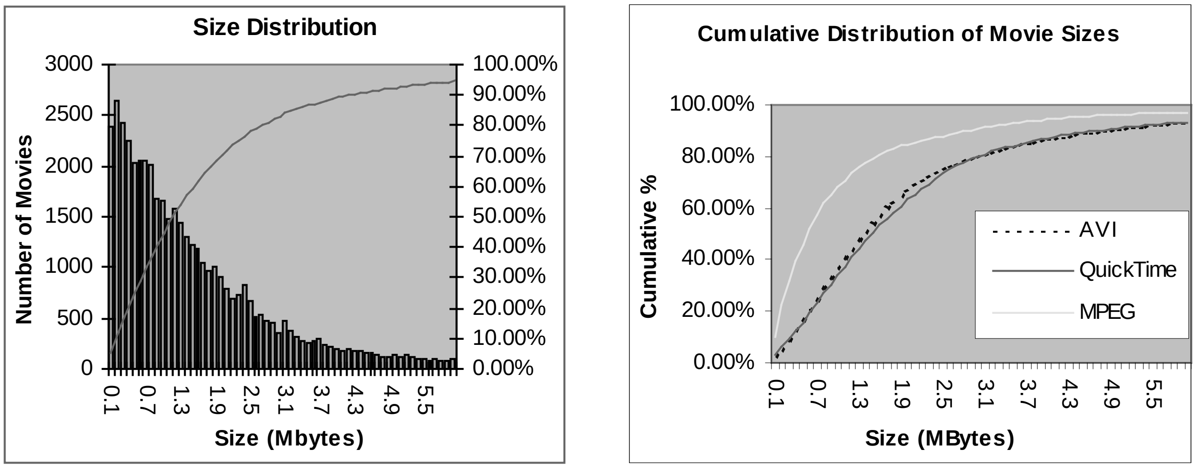


Figure 5: Size distribution of video files (left) Size distribution by type (right)

popularity.

3.2 Frame Rate

AVI and QuickTime movies use low frame rates. Figure 4 displays the frame rate spectrum for all the movies in our data set. At the low end of the spectrum, there are peaks at 8, 10, 12 and 15. At the high end, most fps values cluster around 30 with smaller peaks at 24 and 25. The lower valued peaks in figure 5 are caused by QuickTime and AVI files, while the high peaks are largely due to MPEG files.

3.3 Size

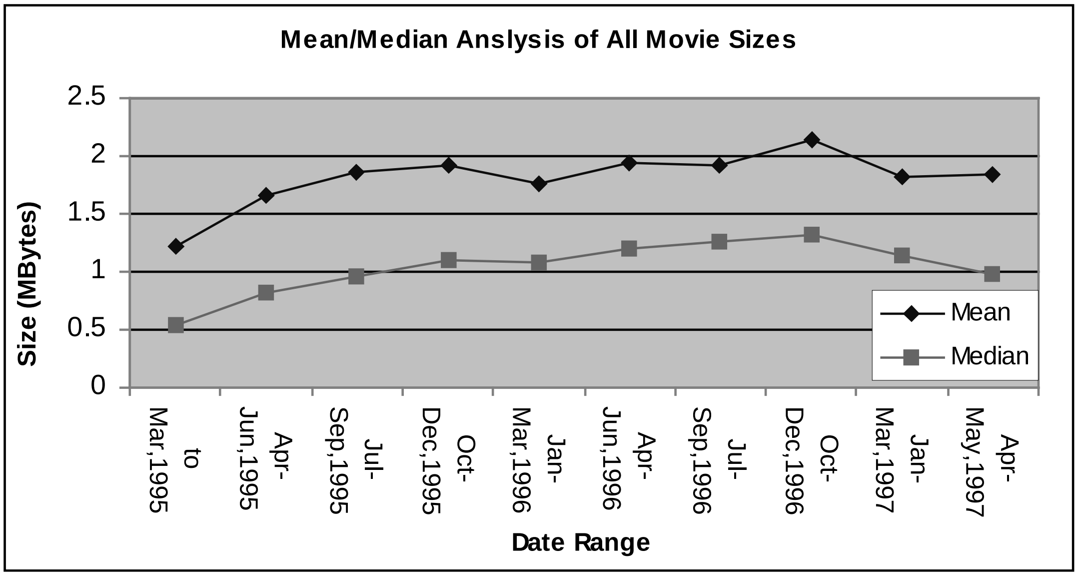
Movie files are relatively small. Figure 5 plots the size distribution of all movies in our data set on the left. It shows that 70% of the movies are 2 Mbytes or less. The median movie size is about 1.1 MB. The right side of the figure breaks this distribution down by format. AVI and QuickTime files have similar size distributions, whereas MPEG files are smaller overall. As we shall see later, this characteristic of MPEG files can be attributed to their better compression and relatively smaller playback times.

3.4 Monthly Size

The median size of the typical movie is increasing, but the median size of a movie file of a given format is staying the same. To see how this seemingly contradictory statement is true, examine figure 6, which plots the mean and median size of movie files versus time across all movies (top) and by movie type (bottom). Each data point represents one three-month period, except the first point, which represents a fifteen-month period. The expansion of the first time segment was required to provide sufficient data points. As the top figure shows, the median size is clearly increasing. The bottom figure just as clearly shows that the median size of MPEG and QuickTime movies is remaining constant. The reason for the rise in the top figure is the increase in popularity of QuickTime (figure 3). Since QuickTime movies are generally two to three times larger (in bytes) than MPEG movies, as shown in the lower figure, a high percentage of QuickTime movies drives the average up. The drop in the popularity of QuickTime in 1997 accounts for the decline in the last two quarters of the median and mean movie size in the top graph.

3.5 Duration

Movies on the Web are short. We calculated the duration of a movie by dividing the number of frames in the movie by the frame rate. Figure 7 (left) shows the number of movies of a given duration, and figure 7 (right) breaks down the left figure by format. 90% of movies are 45 seconds or less in duration, and half of the movies were fifteen seconds and under. The right-hand figure highlights another interesting result: MPEG files are generally shorter than their AVI/QuickTime counterparts. In fact, the latter two formats have almost identical duration distributions.



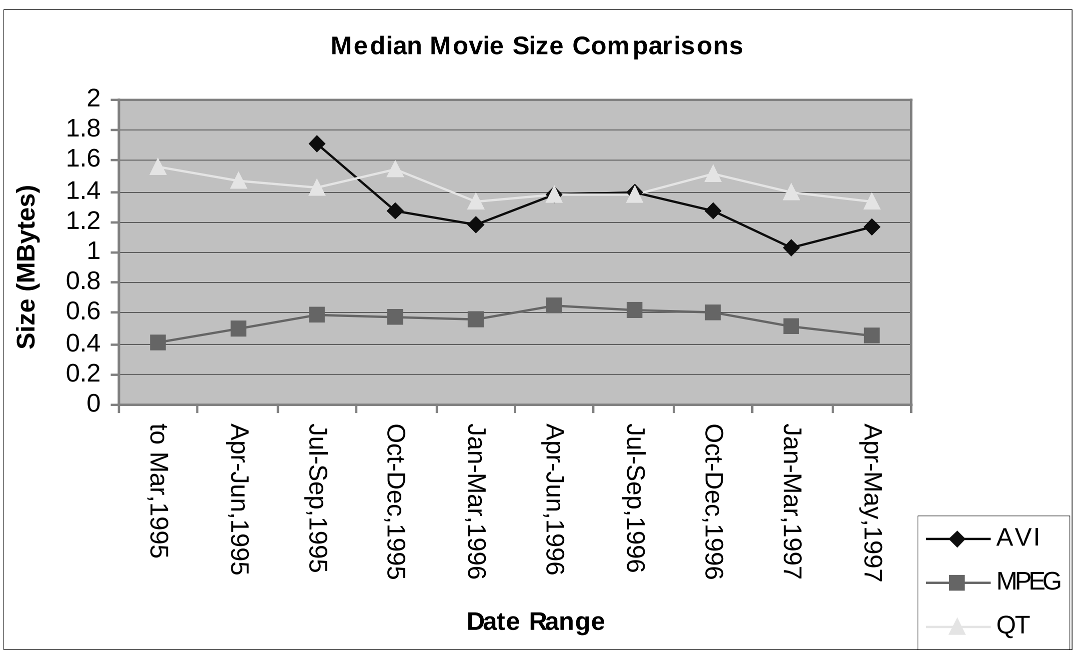


Figure 6: How movie size is changing

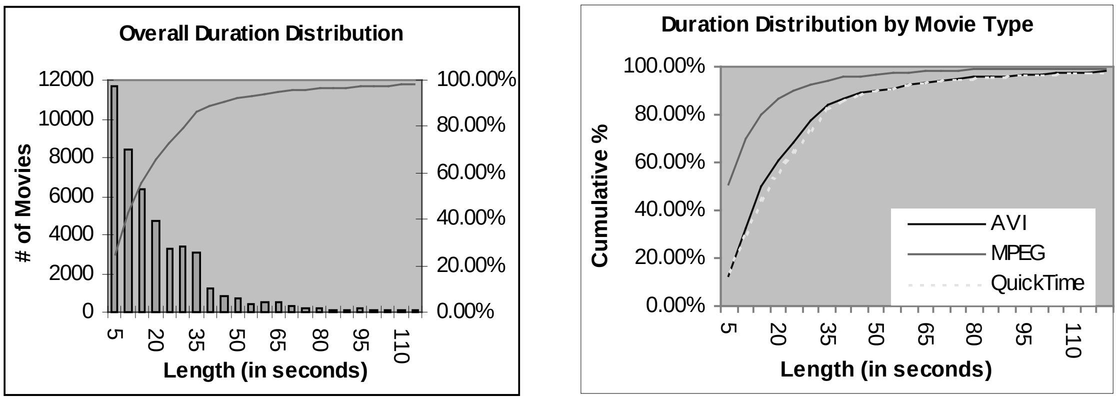


Figure 7: Duration of all movies (left), and duration of movies by type (right)

3.6 Aspect Ratio

74% of all files have an aspect ratio (width over height) of 1.33, which corresponds to a movie size 160x120 and 320x240. 15% of the remaining files have aspect ratios ranging in between 1.2 and 1.5.

3.7 AVI

About 25% of all the AVI files had no audio. 90% of the audio/video files used PCM as their audio codec. Radius Cinepak was the most popular video codec (43%), followed by Microsoft Video 1 (26%) and Intel Indeo R3.2 (25%).

We used the bits/pixel metric to analyze video compression performance:

bits/pixel = video size (bits)/ (width height* number of frames)*

We computed the metric on video-only files and figure 8(left) displays the resulting distribution. The mean bits/pixel was 2.51 and the median was 2.14. Both Radius Cinepak and Indeo had similar mean bits/pixel performances at around 2.0 bits/pixel and Microsoft Video was slightly worse at 2.4 bits/pixel.

3.8 QuickTime

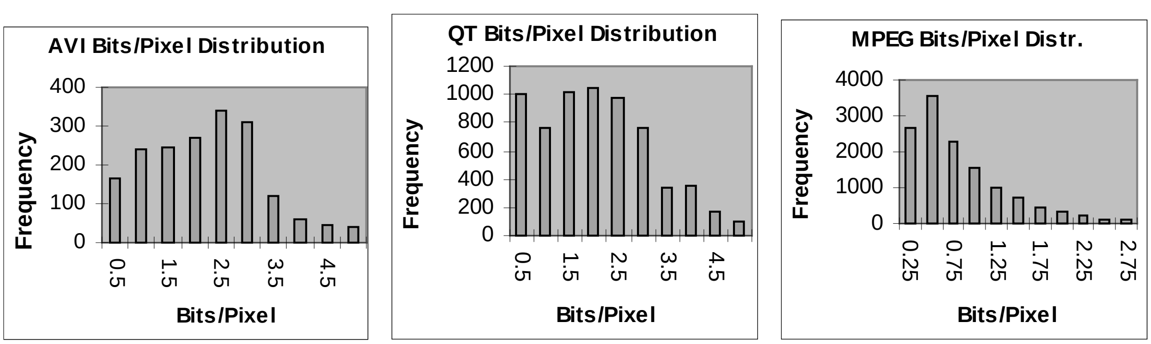


Figure 8: Typical compression performance of formats

About a third of the QuickTime files were video only. PCM was again the dominant audio codec for audio/video streams (84% of all the a/v QuickTime files). Figure 8(center) details the overall bits/pixel distribution. Although it is similar to AVI, QuickTime compresses slightly better with a mean bits/pixel value of 2.16 and median of 1.82.

The most popular video codecs were Radius Cinepak (60%) with a median bits/pixel of 1.9 and Apple Video-RPZA (22%) with 2.6 bits/pixel. We found the best video compression to come from the JPEG codec (6% popularity) which had a median bits/pixel of 1.6.

3.9 MPEG

Figure 8(right) illustrates that MPEG's compression is superior to that of QuickTime or AVI, since the bits/pixel distribution is more concentrated in the low bits/pixel range. We found the MPEG files to have a mean bits/pixel value of 0.73 and a median of 0.53. Only 7% of MPEG files had audio, in contrast to QuickTime or AVI.

Table 2 provides the statistics on individual MPEG frame types: P frames compress about twice as much as I frames, and B frames compress by a factor of 5 better than I frames.

Table 2: Frame Type Analysis

Frame Type	Mean bits/pixel	Median bits/pixel
I	1.25	1.10
P	0.76	0.54
B	0.31	0.19

Investigating the frame patterns in MPEG streams showed that about 80% of all MPEGs had some type of repeating frame pattern. Table 3 shows the various patterns and the corresponding mean bits/pixel. The pattern of I frames only recurred most often followed in popularity by the sequence IBPBB. Note that the bits/pixel value drops when more B frames, relative to P and I, are in the pattern,. The presence of common frame patterns indicates that MPEG users are content to use the default values in their encoders.

Table 3: Frame Pattern Distribution

frame pattern	% distribution	Mean bits/pixel
I	27.1%	1.17
IBBBPBBBPBBB	4.4%	0.66
IBBBPBBBPBBBPBBB	2.9%	0.58
IBBPBB	15.7%	0.70
IBBPBBPBBPBB	8.1%	0.50
IIP	3.5%	0.70
IPBB	2.0%	0.62
IPBBBPBBBB	1.9%	0.28
IPBBIBB	4.2%	0.39
IPBBPBBPBBPB	1.2%	0.51
IPPP	1.2%	0.79
IBBPBBPBBPBBPBB	10.4%	0.31

3.10 Bandwidth Requirement

The Internet is incapable of streaming most MPEG, QuickTime, or AVI video stored on the Web. Figure 9 shows the average bitrate distribution, calculated as the movie size (in bytes) divided by its duration (in seconds). As we can see, at 56Kbits/sec only 1% of the titles can be streamed. At 700 Kbits/sec, 52% of all movies can be streamed, and at 1.5 Mbits/sec 84% can be streamed. Clearly, a disparity exists between bitrates achieved by established compression technologies and current modem speeds. We also find it interesting that despite the pressure to make the viewing of MPEG, QuickTime, and AVI video tolerable over slow connections, authors seldom drop below 500 Kbps when creating their content. This suggests that this bitrate represents a lower limit in quality for these codecs. Below this bitrate, the quality is simply unacceptable.

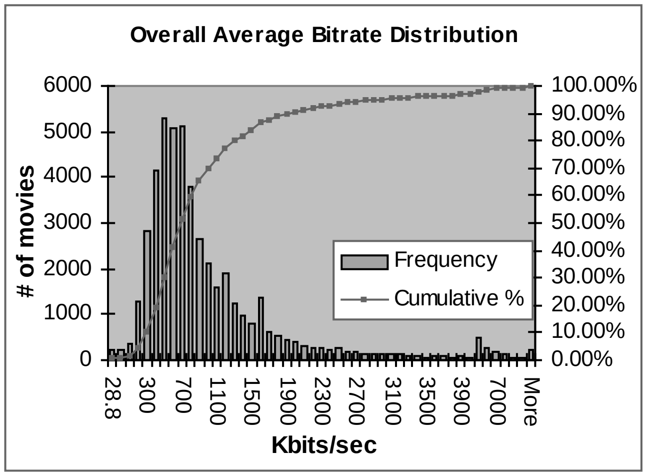


Figure 9: Bitrate requirements for streaming video

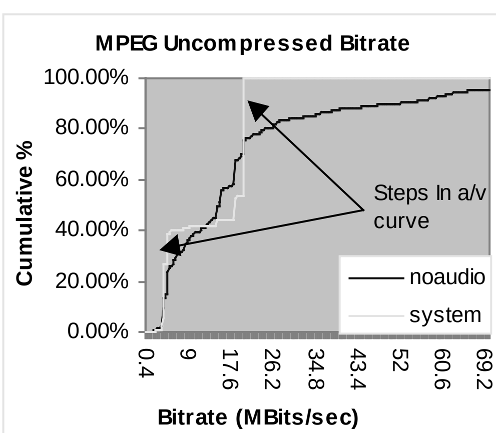
To investigate bitrate distribution further, we defined the property of *transferability*: a file is transferable at a certain bandwidth if its average bitrate is at or below that bandwidth. We first classified our movie collection by type and, within each type, subdivided further depending on whether the movie was video-only or had both audio and video. We then calculated the transferability, at various bandwidths, of the files in each category. Table 1 itemizes the results.

Table 1: Comparison of BW

	QT	QT w.o. Audio	AVI	AVI w.o. audio	MPEG	MPEG w.o. Audio
28,800	0%	2.70%	0%	0.20%	0%	0.06%
56,000	0%	5.55%	0%	0.99%	0%	0.19%
200.000	0.69%	12.26%	1.44%	9.34%	1%	6.35%
600,000	43.28%	42.39%	41.58%	38.22%	36%	37.58%
1,500,000	91%	79.02%	84.73%	78.28%	87%	75.51%
5,000,000	99.77%	96.67%	99.65%	97.42%	97.30%	95.43%
10,000,000	99.97%	99.33%	99.95%	99.25%	99.20%	98.91%

We observe two main points:

1. QuickTime was generally more transferable than either AVI or MPEG, and MPEG was the least transferable. We hypothesize that this is due to its high frame rates, which raise its average bitrate.



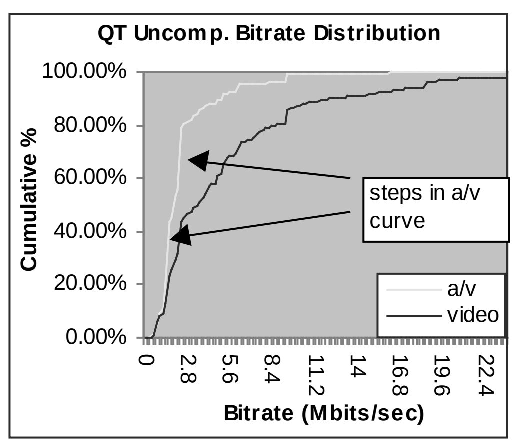


Figure 10: Uncompressed bitrate for MPEG (left) and QuickTime (right)

2. Within each format, audio/video streams are *more* transferable than their video only counterparts at the higher bandwidths. At first, we found this effect counter-intuitive - we expected the presence of audio to raise the average bitrate, not lower it.

To investigate the cause of observation 2, we plotted *uncompressed bitrate* of each video stream:

$$U = 8 * video\ width * height * fps$$

Figure 10 (left) plots the cumulative distribution of U for MPEG video-only and audio/video files. It shows, for examples, that 40% of MPEG audio/video files have $U \leq 2$ Mbits/sec, and almost all have $U \leq 5$ Mbits/sec. Figure 10(right) plots the same metric for QuickTime. The steps in the audio/video curves are caused by files created with common picture dimensions and frame rates. For example, the MPEG systems file curve in figure 16(a) has a large step at the 5 Mbits/sec region and another around the 20 Mbits/sec region. The magnitude of the steps indicates that the majority of the files are located around these regions. The first step is due to 160x120 files (30 or 25 fps), and the second step is caused by files 352 x240 in dimension (30 fps) and 352x288 in dimension (25 fps). The plot of QuickTime files shows a similar pattern. Here the steps in U for audio/video files occur around the 2.4 bits/sec (160*120, 15fps), 2 Mbits/sec (160*120, 12 fps) and 1.6 Mbits/sec (160*120, 10 fps) regions. The AVI uncompressed bitrate distribution is very similar to that of QuickTime, and therefore not shown.

In contrast, video-only files have no such strong characteristics. They have a large variety of shapes and sizes, as reflected in the distribution of U shown in figure 10. Spot checking of the video files indicates that this is because video-only movie files are often used to present the output of simulations and computer animations, which do not have the size restrictions of NTSC or PAL video, the typical source for audio/video data.

3.11: Movie Replication

Our criteria for considering a movie to be a copy of another was as follows: if a movie on a *different* WWW server had the same size, type, width, height and fps we considered the movie to be replicated. Although this does not directly compare the two movies, it is similar to comparing a checksum. Random testing showed our criteria reliable. We found 2177 unique movies that had been replicated. The majority of movies were replicated once with the maximum being 53. Popular replicated movies included “standard” reference files such as “bike.mpg”, “moglie.mpg” and “RedsNightmare.mpg”.

During the gathering phase of our survey, we were careful to check URLs for duplicates to avoid unnecessary processing and downloading. However, this particular analysis revealed a potential gray area: how to differentiate between movies present on the same WWW server with identical replication characteristics but with *different* path names? For future analyses, we plan to ignore such duplicate instances.

4. Related Work

Surveys dealing with the WWW can be classified into roughly two types: examination of Web traffic and analysis of web content. Web traffic investigation can deal with requests either emanating from a cluster of clients or directly at the server itself. Mogul [14] and Kwan [11] have investigated access patterns at specific servers. In addition to analyzing the underlying systems and network behavior of the server under study, they have also examined incoming HTTP requests by looking at their interarrival times, variations with time, size and type of files desired and requesting domain type. The same core criteria (plus some others) have been used by Arlitt [13] to extract underlying patterns from a number of server traces. Cunha et al [12] have performed client side traffic work. They instrumented browsers at clusters of workstations to collect individual user access traces, which they then collated and analyzed. In all of these studies, videos accounted for a very small percentage of overall requests. However, since the traffic data in these studies were all collected during 1994 and 1995 when the web presence of videos, as our study shows, was insignificant, they do not present an accurate picture of current video activity.

Woodruff [16] and Bray [17] have inspected WWW content. They have looked at a very large number of URLs and HTML documents to characterize document sizes, HTML tag usage, file types used as URLs and so on. Their results indicate that videos do not account for a very significant portion of WWW content. Once again, their analysis is based on data collected during 1995. However, according to Woodruff, MPEGs comprise the highest proportion of video files during this time, followed by QuickTime and AVI respectively. This result agrees with our findings.

The study by Smith and Chang [15] is, to our knowledge, the only previous work that has analyzed video on the web. They have implemented a system for traversing the web that locates and indexes images and videos. Their focus is on marrying text-based processing and content-based visual analysis to produce an easily searched taxonomy. Videos form a small portion of the data they have gathered. Our approach concentrates on the direct analysis of videos and how they integrate into the World Wide Web. Our data is unique since it is the first large-scale study of this type.

5. Conclusion

In this paper, we reported the results from an experiment where we collected, characterized, and analyzed a large number of video files on the Web. Our key findings are summarized in section 1. Since network congestion and modem bandwidths are hindrances for wide dissemination of video, it is interesting to observe how user behavior and video technologies have evolved to address these problems. It is clear existing compression technologies do not provide low enough bitrates for streaming transmission over standard modems. One option is to raise the network bandwidth, and our study indicates that 700 Kbits/sec to 1.5 Mbits/sec is an appropriate value. Another approach is improved video[3] compression technologies [18-20] that reduce the required bandwidth. Users of MPEG, AVI or QuickTime are attempting to reduce their bandwidth requirements by creating files relatively small in size and duration (when compared with their VHS counterparts). However, authors have not throttled the bitrate of the videos at the expense of picture quality although, typically, encoders do provide that option. This implies they are not willing to sacrifice video quality for bandwidth - there is a perceptual threshold below which authors are unwilling to descend. The corollary of this is that every video technology has some sort of critical bandwidth associated with it - users cannot tolerate the picture quality for videos encoded below this bandwidth. It may be that the newer video codecs have lower critical bandwidths than those we have investigated, but we are not sure they represent the correct solution since their encoding schemes are proprietary. Additionally, our report indicates that MPEG, AVI and QuickTime have a large user base that will be an obstacle for any other technology.

In the future, we plan on further investigation of video movie distribution over October 1996 – March 1997. We also aim to test how videos age by revisiting each title we have analyzed in the past.

References

1. J. Ousterhout, H.L. DaCosta, D. Harrison, J. Kunze, M. Kupfer, and J. Thompson, *A Trace-Driven Analysis of the Unix 4.2 BSD File System*, in Proc. 10th SOSP, December 1985.
2. Mary G. Baker, John H. Hartman, Michael D. Kupfer, Ken W. Shirriff, and John K. Ousterhout, *Measurements of a Distributed File System*, in Proc. 13th SOSP, Oct. 1991.
3. http://www.altavista.digital.com/
4. http://www.sunscript.com
5. http://www.rahul.net/jfm/avi.html
6. http://quicktime.apple.com
7. D. Le Gall, *MPEG: A Video Compression Standard for Multimedia Applications*, Communications of the ACM, Volume 34, No 4, pp. 46-58, April 1991.
8. http://www.cs.cornell.edu/Info/Projects/zeno/Projects/Tcl-DP.html
9. ftp://bmrc.berkeley.edu/pub/mpeg/stat/
10. http://xanim.va.pubnix.com/home.html
11. Thomas T. Kwan, Robert E. McGrath, and Daniel A. Reed, *User Access Patterns to NCSA's World Wide Web Server*, CS Tech Report UIUCDCS-R-95-1934, University of Illinois at Urbana-Champaign, February 1995.
12. C.R. Cunha, A. Bestavros, M. E. Crovella, *Characteristics of WWW Client Based Traces*, Technical Report TR-95-010, Computer Science Department, Boston University, July 1995.
13. M. Arlitt, C. Williamson. *Web Server Workload Characterization: The Search for Invariants*, ACM SIGMETRICS 96-5, Philadelphia, PA, USA, 1996.
14. J. C. Mogul, *Network Behavior of a Busy Web Server and its Clients*, DEC WRL Research Report 95/5, October 1995.

---

[3] To confuse matters, many of the companies developing the newer schemes are starting to package their technologies as codecs for AVI and QuickTime. However, currently, these codecs are not popular enough for us to consider them part of either technology.

15. J. R. Smith and S.-F. Chang. *Searching for Images and Videos on the World-Wide Web*, Columbia University CTR Technical Report 459-96-25, August 1996.
16. Allison Woodruff, Paul M. Aoki, Eric Brewer, Paul Gauthier and Lawrence A. Rowe, *An Investigation of Documents from the World Wide Web*, in Proc. 5th International World Wide Web Conference, Paris, France, May 1996, pp 963--979, <URL:http://www5conf.inria.fr/fich_html/papers/P7/Overview.html>.
17. Tim Bray, *Measuring the Web*, in Proc. 4th International World Wide Web Conference, Paris, France, May 1996, pp 994--1005, <URL:http://www5conf.inria.fr/fich_html/papers/P9/Overview.html>.
18. http://www.vxtreme.com
19. http://www.vdolive.com
20. http://www.realmedia.com