Data Compression Standards

Daniel C Markert
Computer Science
University of Minnesota Morris

Abstract

This paper first explores the history of the Digital Video MPEG Standards as created by the MPEG Committee of the International Standards Organization(ISO) for data compression. Included are existing standards, i.e. MPEG-1,2,4, and 7. Additionaly, recent developments to the MPEG-21 standard that is still in production will be looked at. For each of the standards, a brief overview of the standard will be given, followed by a summary of some of their key points, technologies, and abilities. The paper then uses this key information to discuss data compression of different media including video, audio and multi-media applications, Examples will be given to show differences between algorithms used for the data compression. Each section of the paper will cover key concepts and ideas to provide the reader with a suitable background for further research.

Introduction

By introducing powerful standards for the compression, representation, and organization of digital video, the MPEG group, a division of the ISO, made it possible for the widespread use of digital video.

Currently, there are five Digital Video Standards that the MPEG committee has introduced, from MPEG-1 to MPEG-21, which is still in its early stages. The format of the paper is as follows:

1. MPEG-1 (A standard used primarily in the storage of video)
2. MPEG-2 (The standard preferred by DVD, SVCDs)
3. MPEG-4 (Designed with multimedia applications in mind)
4. MPEG-7 (Describes all the different facets of a video)
5. MPEG-21 (Creates a multimedia framework)

For each of the standards, a brief overview of the standard will be given, followed by a summary of some of the key points, technologies, and abilities of each standard. It's impossible to go over each of the standards in great detail, but an effort will be made to provide the reader with a basic understanding of the material.

Figure 1. The lack of change from one frame to the next

MPEG-1

Standardized in 1991, MPEG-1 is the oldest of the digital video standards produced by the MPEG group. The purpose of MPEG-1 was to provide technologies and algorithms for the efficient compression of digital video. This was done to facilitate the transport and storage of digital content, while maintaining quality.

Video content is very redundant because it is made up of a series of still frames. As such, the difference between frames in a given sequence is minimal, if it changes at all. There may also exist redundancies within each frame, i.e. a certain color could be used to represent an entire area of pixels. Figure 1 shows two sequential frames that the author ripped from a DivX video. It shows the lack of change from one frame to the next. The algorithms and technologies of MPEG-1 are designed to take advantage of these redundant elements.

The importance of the techologies and concepts defined by MPEG-1 for the compression and transport of video is that they form the basis for later standards. To cement the basic concepts of MPEG-1, a real-life example of how compression works will be demonstrated, followed by a look at the encoder-decoder block algorithm.

MPEG divides videos into sequences that group together similar frames. A frame is nothing more than a still image. The frames, in each sequence, are then divided into groups of I, P, and B-Frames (explained below). These frames are important in determining how compressed a frame can be, the chance of error propagation, whether it represents inter-frame or intra-frame data, and the encoding order.

If a frame is an I-frame, then it is an intra-coded frame. Being intra-coded means that it looks for redundacies only within its own frame. Because it is self-referencing, I-frames are always used to start a video-sequence. Also called the "key-frame," an I-frame is the frame that media players point to when moving through a videoSikora, T.Ebrahimi and M.Kunt.

The P-Frame (Predictive) uses information found in previous frames and then performs motion compensation (MC) to "guess" what its values are. P-frames achieve a higher degree of compression compared to I-Frames, but can be risky since they can be used to reference other P-Frames, which may cause errors to propagate.

B-Frames (Bi-directional) are like the P-Frames, but look at future frames as well as previous frames to construct themselves. The B-Frame achieves even higher compression than a P-Frame and doesn't have propagation errors because it's not used as a reference for any other frame.

These 3 frames are then grouped together in many different ways to form a video sequence, which starts off with an I-Frame, and then includes a series of more P, B, and I-frames. Thus sequences like IPPBPP or IPPIIBIIP are valid sequences of frames. Of course, there are some that provide better compression than others, but many sequences are possible.

Once the frame type ratio and ordering is worked out, the video can be encoded. To organize a frame for compression, MPEG-1 follows a divide and conquer approach. Individual frames are cordoned off into a grid of squares called macroblocks. The standard size for intra-coded macro blocks is 8x8 pixels, where each pixel represents a single color. Each of 64 pixels in a macroblock are encoded and compressed and then reconstructed by the decoding process (Watson, 1994; Sikora, 1997, ; T.Ebrahimi & M.Kunt, 1998).

There are three steps in standard MPEG video compression

1. The image is transformed into a simpler representation.
2. The numerical values from step 1 are quantized.
3. Finally the quantized values from step 2 are encoded using entropy.

If done properly, and the encoded video is smaller than the original, then compression has occured. To make the process clearer, lets go over each of the steps in some detail.

The first step in the compression process is to transform the frame's macros into something simpler and smaller. There are a variety for techniques that do this; the one commonly used by MPEG is the Discrete Cosine Transform(DCT) (Sikora, 1997,).

The DCT basically takes a matrix of values representing a macroblock, an 8x8 macro in this case and, using various vectors and cosines, transforms the pixel values into something smaller (Watson, 1994). There are a few different DCT transforms used in image compression. The one used in this paper is as follows: For any size matrix nxm, the value of the transformed pixel at point (u, v) is given by the equation in Table 1.

Table 1: DCT Transformation Equation

$$S(u,v) = \frac{2}{\sqrt{nm}} C(u)C(v) \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} s(x,y) \cos\frac{(2x+1)u\pi}{2n} \cos\frac{(2y+1)u\pi}{2m} u = 0 \cdots n, v = 0 \cdots m$$

$$C(u) \text{ and } C(v) = 2^{-1/2} \text{ if } u, v = 0 \text{ else } 1 \quad (1)$$

$$s(x,y) = \text{ the pixel value} \quad (2)$$

What effect does this equation have on a macro? Figure 2 shows the matrix before

$$\begin{bmatrix} 40 & 120 & 131 & 107 & 133 & 142 & 132 & 148 \\ 70 & 148 & 160 & 136 & 159 & 168 & 156 & 173 \\ 103 & 182 & 191 & 166 & 187 & 194 & 181 & 196 \\ 117 & 194 & 201 & 174 & 192 & 197 & 181 & 196 \\ 113 & 189 & 194 & 163 & 178 & 180 & 163 & 175 \\ 109 & 184 & 187 & 154 & 167 & 166 & 147 & 159 \\ 119 & 193 & 194 & 159 & 170 & 167 & 147 & 156 \\ 131 & 205 & 206 & 168 & 179 & 174 & 154 & 164 \end{bmatrix}$$

$$\begin{bmatrix} 255 & 19 & 13 & 8 & 2 & 0 & 70 & 70 \\ 15 & 15 & 70 & 69 & 70 & 70 & 70 & 70 \\ 6 & 70 & 70 & 70 & 69 & 70 & 70 & 71 \\ 7 & 70 & 70 & 70 & 70 & 70 & 70 & 70 \\ 69 & 70 & 70 & 70 & 70 & 70 & 70 & 70 \\ 70 & 70 & 70 & 70 & 70 & 70 & 69 & 70 \\ 70 & 69 & 70 & 70 & 70 & 69 & 69 & 70 \\ 70 & 71 & 70 & 70 & 69 & 70 & 69 & 70 \end{bmatrix}$$

Figure 2. matrix array showing converted the values of the macroblock in Figure 2 before and after DCT

and after the DCT process. The DCT matrix contains values that are smaller and not as varied as the original. The exception to this is the value at $(0,0)$. The DCT of this pixel is always the largest because it is computed using every pixel in the macro. Something else to notice is that the less frequent values appear in the upper left, while the more frequent ones appear in the lower right. This occurs because in macros the most important values, i.e. the ones that are most noticible, are usually in the upper left. The DCT transform reflects this. (Watson, 1994)

After the DCT is performed, the next step in the compression process is to quantify the 8x8 DCT matrix of coefficients. This is necessary because while the DCT process has helped group together the most important pixels, quantification can simplify things even further (Watson, 1994; Sikora, 1997,).

Quantification is the process of finding some method, description, model, etc $\cdots$ of describing a set up values in order to lower the total number of values that need to be represented. This allows us to shrink the total number of bits necessary to represent information (Watson, 1994).

MPEG-1 quantifies data by dividing it with a certain weight. For example, the values of the DCT Matrix in Figure 3 could all be divided by a weight of 64 and then rounded to the nearest integer. Dividing values by a weight is preferred, as opposed to just rounding, because it raises the total number of values possible, which makes finer details show up better. After being divided by the weights, the values are then rounded to the nearest integer because most quantified integers can be stored with only 3 bits as opposed to floating point values and large integers which the DCT often gives us (Watson, 1994;

Sikora, 1997,).

The method MPEG-1 uses to quantify data is to divide the 8x8 intra-matrix by a corrosponding 8x8 matrix of weighted values. These are not uniform values and use lower weights for more important values and higher weights for pixels that don't matter much. Looking at Figure 3, what stands out more: the bright whitish pixels, or the greyish blobs of higher frequency values? In most cases, the answer would be the white pixel because it is not as frequent and therefore stands out more (Watson, 1994; Sikora, 1997,).

One of the weighted matrices that MPEG uses to quantize intra-coded macro-blocks is shown below.

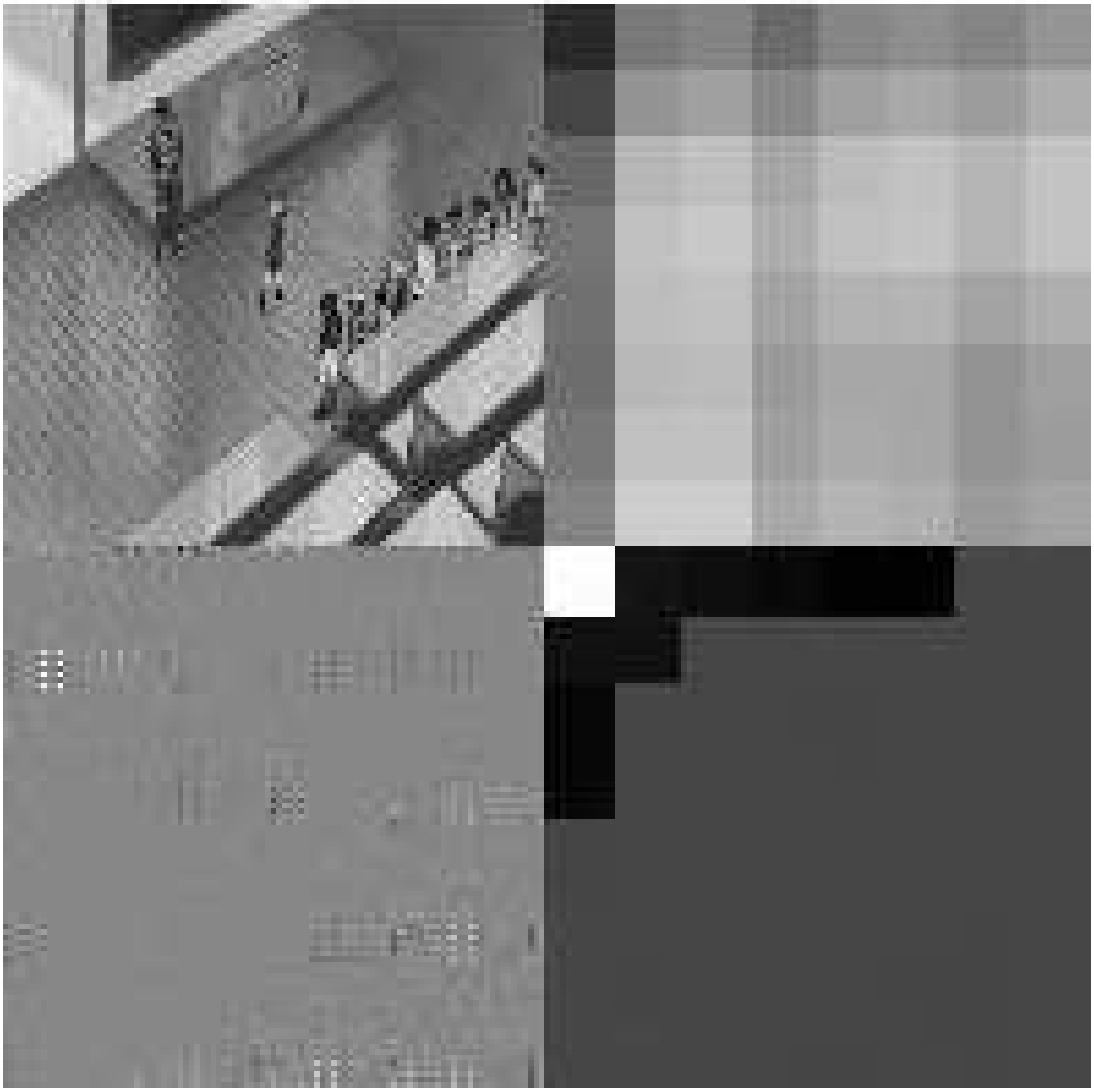

Figure 3. Showing Frame and Macro before and after DCT. (Ripped and Converted by the Author)

$$\begin{bmatrix} 8 & 17 & 18 & 19 & 21 & 23 & 25 & 27 \\ 17 & 18 & 19 & 21 & 23 & 25 & 27 & 28 \\ 20 & 21 & 22 & 23 & 24 & 26 & 28 & 30 \\ 21 & 22 & 23 & 24 & 26 & 28 & 30 & 32 \\ 22 & 23 & 24 & 26 & 28 & 30 & 32 & 35 \\ 23 & 24 & 26 & 28 & 30 & 32 & 35 & 38 \\ 25 & 26 & 28 & 30 & 32 & 35 & 38 & 41 \\ 27 & 28 & 30 & 32 & 35 & 38 & 41 & 45 \end{bmatrix}$$

After the macro in Figure 2 is quantized, the resulting maxtrix of quantized values becomes:

$$\begin{bmatrix} 32 & 1 & 1 & 0 & 0 & 0 & 3 & 3 \\ 1 & 1 & 4 & 3 & 3 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 & 3 & 3 & 3 & 2 \\ 0 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

The value at $(0, 0)$ stands out amongst the rest, and that's what we wanted. The the other values have been shrunk so that for any given pixel, the value can only be 1 of 5 possible values. This makes it easy to see the redundacy in the array, and now that the values are in integers, it can be entropy-coded with favorable results.

Entropy-coding, in the computing sense, finds alternative ways of representing the binary stream of a set of data, as opposed to a straight 1:1 conversion, like ASCII coding. A popular method of entropy-coding, and one of the kinds used for image compression, is Huffman-Coding (Astrachan,).

Huffman Coding is a way of compressing a set of data. What it does is look at the data and construct a Huffman Table. This table charts every character in the data and how often it occurs. From this table, an optimal-tree is created, which maps the least number of bits to the characters appearing most. For our Q-Matrix, the Huffman Table is a simple one, and is shown in Table 2 with the corrospoding variable length code(VLC) received from the optimal tree (Astrachan,).

The pixels are then encoded into a bitstream based on the VLC Table above, along with headers that contain the information necessary for successful decoding of the compressed bitstream (Astrachan,).

All the steps that we have just gone through: the transforming of a macroblock, the quantizing of its values, and the subsequent entropy-coding are repeated for every macro in every frame of the video, and are combined into a bitstream that's passed along in chunks to the decoder. That bitstream that we finally arrive at is smaller than what would have resulted had we used ASCII/UNICODE Tables. Because of that, the video is considered to be compressed (Astrachan,).

Table 2: Variable Length Huffman Table

Char	Freq	VLC
2	34	00
3	19	01
0	5	100
1	4	101
4	1	110
32	1	111

Now that the small picture has been looked at, lets look at the big picture by examining the Encoder-Decoder Diagram. This diagram, shown in Figure 4, plots out the process a video file goes through from the encoding, designed to compress the video file, to the decoder, which decodes the compressed bitstream at a certain bitrate for viewing purposes.

The Encoder-Decoder Algorithm for MPEG-1 is very simple. If the frame is an I-Frame, if goes through the DCT transform (DCT), is Quantized (Q), and then Variable Length Coded (VLC). It then passes through the Video Buffer (VB) to ensure a constant bitrate is maintained, and because the I-frame may be used for predictive coding of P or B-frames. After the frame is Quantized, it goes through de-quantification (Q^*), and an invserse DCT (DCT^{-1}) is applied to so the original frame is reconstructed. That frame is then placed in Frame Store(FS), so that future P and B-frames can use it for Motion

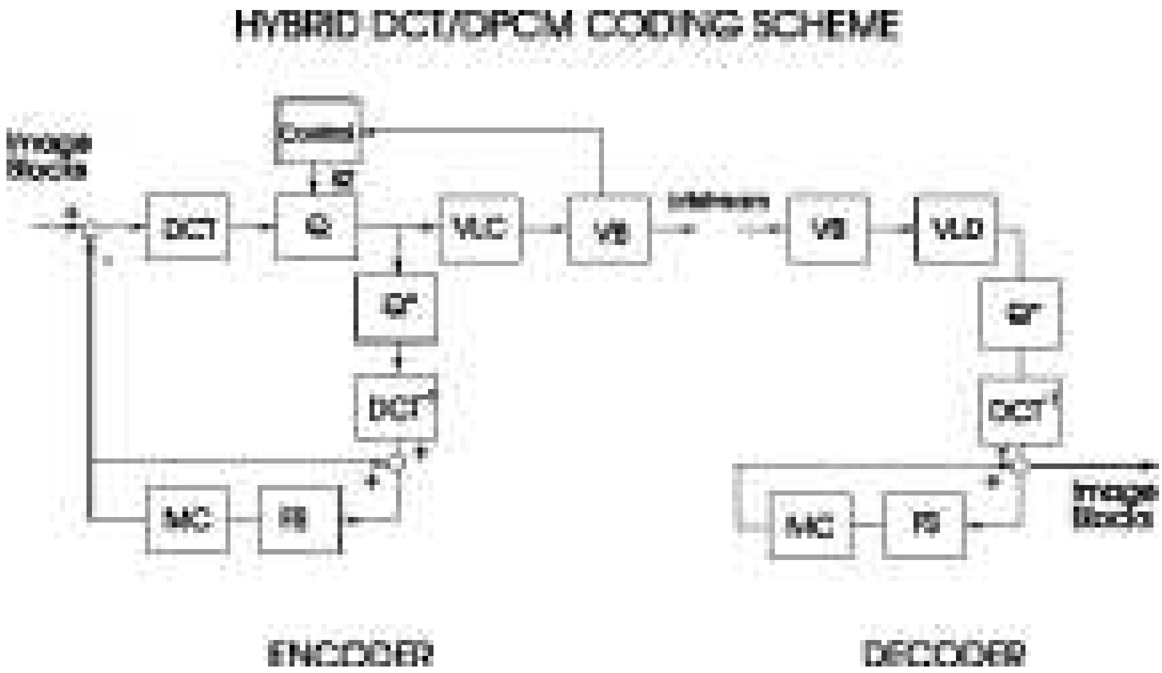

Figure 4. Encoder-Decoder Algorithm[5]

Compensation(MC). The same general process is used for P and B-frames, except B-frames can't be referenced, so only MC is used and the frames are not kept in FS. This process is looped until the frames are converted into a bitstream (Sikora, 1997, ; T.Ebrahimi & M.Kunt, 1998).

The opposite process occurs at the decoder. Here, after the VB, a Variable Length Decode (VLD) is performed, to translate the entropied information back into a matrix. Q^* is then performed on the the macroblocks, which then go through an DCT^{-1}. And like the encoder, for the decoding of predictive frames, I and P-frames are kept in FS until such time as they're needed for MC. This process results in completed frames that can be viewed by media players, etc... (T.Ebrahimi & M.Kunt, 1998; Sikora, 1997,).

As we have seen, compressing video according to the MPEG-1 specification is relatively simple. Yet, it's also a powerful tool because it enables videos to be compressed and yet maintain decent quality for viewing and storage.

MPEG-2

MPEG-2, which became a standard in 1995, is a higher quality version of MPEG-1. It was designed to be backward compatible with formats following the MPEG-1 standard, and uses the same techniques that MPEG-1 does, like DCT, Waveform, and Entropy-based encoding. The difference in quality has to do with the tweaking that many of the technologies went through in order to produce higher bit rates and an overall higher standard of video. Because of its vastly superior quality, MPEG-2 was introduced for a purpose different than MPEG-1. Rather than be used primarily for efficient storage and transport, MPEG-2 is used for the production and displaying of high-quality video content like what satellite, digital, and HD televisions display. Additionally, it is used as the backbone for the DVD format. Though not as popular as DVDs, SVCDs (Super Video CDs) also use MPEG-2 as its backbone component (Sikora, 1997, ; T.Ebrahimi & M.Kunt, 1998).

MPEG-4

Creating and thorough editing of video content was not a major concern of MPEG-1 and -2. MPEG-1 especially was hard to work with using only low-level editing. There wasn't very much an encoder could do to clean up a video. People recognized that MPEG-1 and 2 were not adequate for representing multimedia, and therefore MPEG-4 was created. The first version of MPEG-4 came out in 1995 and it provides a way to create digital video content that would allow for extensive and throrough editing and cleaning. Size was also an important part of the standard. Because of its high compression ratios and the ability to clean up a lot of the artifacts (visible errors) caused by high compression, MPEG-4 content was ideal for computer usersKoenen.

MPEG-4 is very functional. It can do many things with a great deal of information. This section will look at three of the major functionality sections that MPEG-4 defines. These are the Systems, Audio, and Visual functionalitiesKoenen.

MPEG-4 Systems addresses how to describe the relationship between audio and visual objects within digital content. These relationships are either on a spatial-temporal level, or an object-to-object levelKoenen.

The spatial-temporal level helps in describing how visual and audio objects in a scene are related to each other in a space-time context. To do this, MPEG-4 uses the Binary Format for Scenes (BIFS). This tool allows each object to interact with other objects if so desired. This is ideal for many multimedia applications like those that run an event-script when a user-triggered event happensKoenen.

MPEG-4 divides its audio coding tools into two categories: the Natural codings tools, and the Synthetic tools. The Natural tools deal with low-bitrate audio like speech and high quality multi-channel audio. The Natural tools also help in the compression and transmission of audio. The Synthetic tools provide representations of audio through parametric descriptions. In order to integrate these two sets of tools, MPEG-4 specifies a system called AudioBIFS (ABIFS).

ABIFS is a layer of BIFS and works to construct audio objects using the Synthetic and Natural coding tools of MPEG-4. After the objects are constructed, they are then used by BIFS when it constructs the AV Scene.

ABIFS coordinates and maps the different audio objects together via a scene graph. The scene graph simply lays out the relationship and nature of each audio object for use by the BIFS scene graph higher up. In MPEG-4 version 1, there are 8 different kinds of objects/nodes that an object can be defined as. These nodes contain fields for the definition of the object, as well as parameter sets that the object might need for operation. See Table 3 for definitions of the 8 audio nodes defined in MPEG-4 version 1. To see an example of how the nodes work, lets look at the AudioFX node.

The AudioFX node applies different custom effects across several channels of input. It does this through the SAOL processing language. This allows audio to be mixed with artifical input like electronic sound effects or gunshots and then outputted to the appropriate channels. Beyond SAOL, SASL (Structured Audio Score Language) is also used. SASL is a language for detailing the behavior of sound along the time axis, which allows for variations of SAOL-produced sound over time, instead of maintaining a constant factor.

Name	Purpose
AudioSource	provides a link from the low-level sound decoders to the scene graph
AudioMix	turns an input of x channels into y channels
AudioSwitch	Selectively chooses from a subset of m input channels
AudioDelay	delays sound for syncronization purposes
AudioFX	adds signal processing effects
AudioBuffer	stores sound for use in interactive environments
Sound	arranges sound in a 3-D environment
Sound2D	arranges sound for use in a 2-D environment
Group	gathers together different nodes for use in heirarchical transformations
Listening Point	Gives the location of the virtual listener in a given scene
TermCap	Finds out what playback resources are available

Table 3: MPEG-4 Version 1 audio nodes

Object Descriptors(OD) is the second section of Systems. ODs help to describe how video and audio streams are related to each other in a scene. ODs can also be used for Internet streams. For example, you have a talking speaker that stands next to a video screen that runs a slideshow. BIFS would examine where the speaker and screen exisited spatially, and would also handle the video and audio streams. The two streams would have ODs attached to them to describe which object they belonged to, where they fit, etc..., and would also contain the URLs of the streams if they needed to be downloaded from a serverKoenen.

The audio section of MPEG-4 supports a wide variety of audio objects like General Audio bits, which range from very simple mono audio tracks to much larger and higher quality multi-channel audio. Others would be objects like Speech Signals that support audio manipulations like pitch variance and volume. Additionally, synthetic audio is dealt with. Synthetic audio is mathematical representations of natural sound. For example, you could have the synthetic sound of a piano and drum mixed together with the sounds of running water over in a creek with the caw of a bird. Lastly, MPEG-4 allows for Synthetic Scalable Speech, which combines a series of parameters (tone, level, accent, pitch, etc.) with text to create viable synthetic sound(Koenen, 2002).

The Visual Functionalities of MPEG-4 deal with a great many kinds of visual information. It includes specs as to what kinds of formats it can support with a given set of attributes like bit rate and resolution. Tools for efficient compression of visual information are also included. Most of what is included, though, has to do with attributes associated with moving videos. It includes definitions for random access features like fast forward, reverse, stop, and seek. Powerful manipulations of content within a video scene is also supported. This includes tools to support interaction between moving objects, text, images, and artificial textures, which are interleaved to form a single sceneKoenen.

In short, MPEG-4 is a powerful tool which, when used to its utmost, can help display information in new and interesting ways and also enable powerful editing of videos.

MPEG-7

The Internet is a powerful tool for looking up information partly due to the power and convenience of search engines. Sites like Google and Lycos enable users to enter a string and then search websites for that string and return hits. This enables users to find information at a far quicker rate than ordinary searching would. The problem is that the current form of searching is text based and is inadequate when searching for multimedia. MPEG-7 addresses this lack of multimedia searching (Martinez, 2002).

MPEG-7, version 1, which was released is a standard designed to provide content that is associated with a set of descriptors. These provide a great deal of information about the Audio Visual(AV) content for fast searching and retrieval by a variety of tools including search engines, databases, and other multimedia applicationsMartinez.

The power of this standard is that it provides a very powerful tool for multimedia searching, and could simplify what might have been nearly impossible with ordinary text-based searches. For example, say I had a clip of audio. I have no idea who the artist is, what soundtrack, or CD it is from. If I wanted to try and find the origin of the clip, the amount of work required would be enormous. However, with the MPEG-7 standard, searching for the artist/CD would be a simple process. I would enter the music clip in a multimedia search engine, and by examining similarities between search fields and descriptors, would be able to find and return things like the full version of the song, information about that song, the artist, and where I could buy the CD.

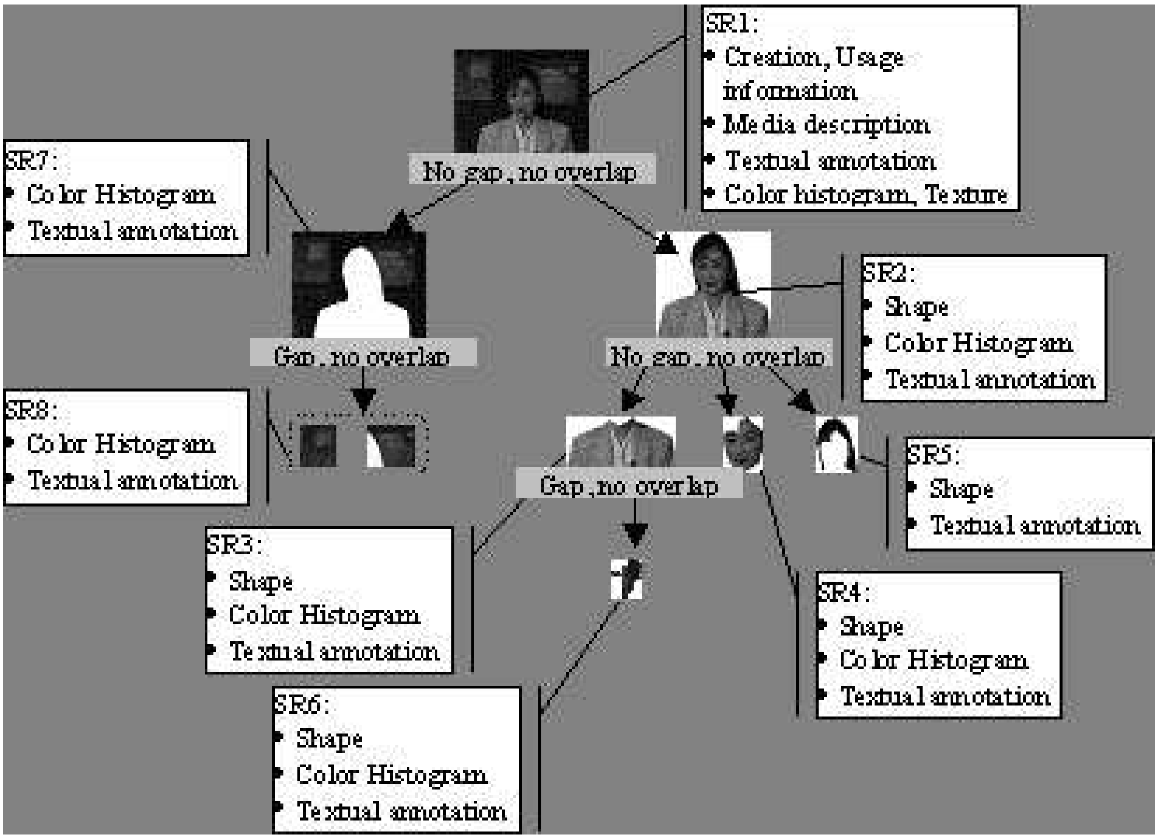

Figure 5. A frame divided up into parts with attached descriptors

MPEG-7 is designed to allow quick searching and retrieving of desired information. For this to happen MPEG-7 content is described using a set of Descriptors (D). These descriptors are words that describe certain aspects about the AV content as a whole as well as things about the individual objects that it's made out of. These descriptors can range from being very vague and abstract; i.e. using the descriptors "round" and "shiny" to describe a shape you're searching for, to very specific searches: "A small round, shiny ball that is 3 inches in diameter." These descriptors are then associated with the content through the use of XML meta-tags. See Figure 5 for an example of how a frame could be divided up into descriptorsMartinez.

Description Schemes (DS) are another powerful tool that MPEG-7 employs. DSs are packages that contain sets of Descriptors and DSs. They are used to gather together sets of Ds and DSs that make sense for an object and thus can be reused over and over again for instances of the same type of object.(Martinez, 2002)

The Description Definition Language (DDL) provides a tool for defining new sets of Ds and DSs. Like most elements of the description process, it is based on XML. Using this language, users are able to create Ds and DSs other than the ones initially supported. This allows for future innovation and also for the introduction of new media types.(Martinez, 2002)

MPEG-7 Visual consists of different models and descriptors that provide powerful and useful descriptions for visual objects. These descriptors and models fall into a category based on how they're used and for what purpose. Currently, MPEG-7 defines six categories: color, motion, shape, texture, localization, and face recognition (Martinez, 2002).

The texture category, for example, has a set of three descriptors: homogenous texture, edge histogram, and texture browsing. These descriptors are used to define important aspects of the visual object that involve texture in some way. The homogenous descriptor, for example, is used for finding similarities between different textures and retrieving the ones that match, and is typically useful if there is a large number of similar looking textures and patterns that need to be filteredMartinez.

Another descriptor is the motion tranjectory descriptor. It is defined by the standard as being a set of keypoints, like (x, y) that have with them functions to describe the tranjectory of a object as it goes from x to y. This allows for precision in the recognition of certain situations like when someone is trespassing into an area that they're not supposed to, or in discerning certain patterns of movement. For example, it could be used to indentify a certain sequence of events in a football game, like the retrieval of all passes by this quarterback to that receiver from the 10 yrd line to the goalMartinez.

Along with video specs, MPEG-7 also has categories, which relate only to audio objects. These models and descriptors fall into two categories depending on whether they describe "low level" or "high level" audio attributesMartinez.

The low level descriptors, like the visual sets, are grouped into categories. The current version of mpeg-7 has 6 groups of low level descriptors and models. These are: basic, basic spectral, signal parameters, timbral temporal, timbral spectral, and spectral basis.(Martinez, 2002).

For example, the timboral temporal category contains two descriptors: logattacktime and temporalcentroid. These two descriptors are used to describe the temporal content that a audio object possesses. The logattackdescriptor is used to describe how long it takes a signal or sound to go from silence to its maximum amplitude. TemporalCentroid descriptions, on the other hand are used to describe when and where, in a given time frame, the energy of a signal is at its most focused. This could be used to describe such content as a scream that starts off very loud and high-pitched and then gradually fades away.(Martinez, 2002)

The High Level tools for audio description are more powerful, and often times combine descriptors and models from the lower-level, which enables less abstraction and more musical descriptions of audio objects. These tools fall into six categories: audio signature, musical instrument timbre, melody description, general sound recognition and indexing, and spoken contentMartinez.

To see how these tools help describe audio, we will look at the musical instrument timbre category. Here one of the descriptors included is the HarmonicInstrumentTimbre descriptor. This descriptor actually combines several different low-level descriptors. It uses the 4 harmonic timbral spectral descriptors (HarmonicSpectralCentroid, HarmonicSpectralSpread, HarmonicSpectralDeviation, HarmonicSpectralVarioation), which are used to describe the different kinds of harmonics, as well as the relationships between them. The HarmonicSpectralDeviation, for example, is used to see how far a given harmonic "deviates" from a global harmonic. The HarmonicSpectralVariation descriptor, on the other hand, finds correlation between two different samples of a harmonic in a given signal. These descriptors are then combined with the LogAttackTime descriptor to describe harmonics of different instruments like pianos, guitars, etc... (Martinez, 2002).

MPEG-21

The focus of the standards so far has been very specific. Mpeg-1,2,4 dealt with the compression and representation of Audio-Visual (AV) objects for storage and viewing (mpeg-1,2), and multimedia (mpeg-4). Mpeg-7 broke the mold slightly by defining a framework for describing an AV object for easier search and retrieval. But in all cases, the quick and efficient exchange of information, like text and AV objects, relies on complex frameworks of nodes and servers, that may not be on the same level, and able to communicate adequately with each other.

MPEG-21, which is in pre-production, provides a "multimedia framework" that will greatly facilitate the transport and delivery of multimedia content across the complex number of networks and devices in use today. In order for this to happen, many different functionalities need to be created. There needs to be event reporting, so that errors, collisions between packets on a network, and changes of personal information as well as other things like a common identification format for multimedia, and ways to protect content through the transport process. To address these issues and many others, MPEG-21 currently includes five elements for its multimedia frameworkJan Bormans.

Currently, the work on MPEG-21 includes five elements that form the bulk of the standard. These elements are:

1. Digital Item Declaration (DID): Defines what "digital" content is and provides a model for itJan Bormans .

2. Digital Item Identification (DII): Uniquely identifies DIDs through such technologies as IP addressing, and Description Schemes as defined by MPEG-7Jan Bormans.

3. Intellectual Property Management and Protection: Gives content providers tools to manage and protect their Digital Content from abuses like illegal copying and distributionJan Bormans.

4. Rights Data Dictionary (RDD): The RDD is a dictionary for legal terms, documents, case laws, and other pieces of information that the REL usesJan Bormans.

5. Data Item Adaptation: Provides a means to adapt or change digital content for transport across slower media and to destinations that are unable to handle the content at its current stateJan Bormans.

In addition to the items above, which are already a set part of the standard, one of the key future components will be interfaces and protocols for advanced Content Handling and Usage. When in place, it is hoped that they will provide essential and useful utilities that allow for searching, storage and management of Digital Items. One of these utilities is the ability of users to create a preferences file for things like Digital Adaptation so that incoming media would know what that user's capability is and be able to adapt itself accordingly. And for the content providers, MPEG-21 is looking to include an advanced tracking feature that would track and maintain a live record of where a user's content is situated and what restrictions, if any, are still in place on it. This would be ideal in tracking pirated copies of software or digital movies. The final release is also planned to include a section that would define methodologies for the management of user profiles. These utilities would let users edit profiles, track content usage information, also also users to create an avatar(entity) of the themselves that would follow predefined paths and dialogues.

Like with most in their infant stages, the MPEG group has set down a very ambitious amount of features and concepts to include in the final release of MPEG-21. It is doubtful that all of them will make the first release, though. There are some things that are pretty much set in stone, but future definitions of categories like Content Handling and Usage may be parsed down a little. Hopefully, however, most of what they would like to include in the standard makes it because of the power and convenience the final standard brings.

Conclusion

From the mpeg-1 standard created in the early 90s to what's currently being done with mpeg-7 and mpeg-21, the MPEG group has produced, and is continuing to produce, standards that are revolutionizing the ability of users to create, store, edit, search, retrieve, and share Audio-Visual content.

References

Sikora, T. Digital consumer electronics handbook. McGraw-Hill.

Astrachan, O. L. *Huffman coding: A cs2 assignment@http://www.cs.duke.edu/csed/poop/huff/info.*

Jan Bormans, K. H. (2002). *Mpeg-21 overview* (Tech. Rep.). Shanghai: ISO/WG11(MPEG)-Requirements Group.

Koenen, R. (2002). *Mpeg-4 overview* (Tech. Rep.). ISO/WG11(MPEG).

Martinez, J. M. (2002). *Mpeg-7 overview* (Tech. Rep.). Klangenfurt: ISO.

Sikora, T. (1997). Mpeg digital video coding standards. *IEEE Signal Processing Magazine*, *14*(5), 82-100.

T.Ebrahimi, & M.Kunt. (1998). Visual data compression for multimedia applications. *Proceedings of the IEEE*, *86*(6), 1109-1125.

Watson, A. B. (1994). Image compression using the discrete cosine transform. *Mathematica Journal*, *4*(1), 81-85.