Experiment 8  VIDEO COMPRESSION

I  Introduction

In the last experiment, we discussed and experimented with techniques for compression of still images. In this experiment, we will move on to compression of moving images or video. Table 1 shows the raw data rate required by digital video in different formats. We can see that a video sequence is substantially more demanding in terms of memory and/or bandwidth than images. For example, a video in the CCIR601 resolution (equivalent to analog TV resolution) with 4:2:0 color subsampling will take 128 Mbit/s or 16 Mbyte/s. In other words, a 2 hour movie will take 112 Gbytes (that is 1120 zip disks!). A lower quality CIF movie (equivalent to the VHS VCR video resolution or the VCD movies) will take a quarter of this (still close to 300 zip disks!). Obviously, more drastic forms of data compression are required for storing and transmitting video than for images.

<table>
<thead>
<tr>
<th>Video Format</th>
<th>Y Size</th>
<th>Color Sampling</th>
<th>Frame Rate (Hz)</th>
<th>Raw Data Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDTV Over air. cable, satellite, MPEG2 video, 20-45 Mbps</td>
<td>1280x720</td>
<td>4:2:0</td>
<td>24P/30P/60P</td>
<td>265/332/664</td>
</tr>
<tr>
<td>SMPT296M</td>
<td>1920x1080</td>
<td>4:2:0</td>
<td>24P/30P/60I</td>
<td>597/746/746</td>
</tr>
<tr>
<td>SMPTE295M</td>
<td>720x480/576</td>
<td>4:4:4</td>
<td>60I/50I</td>
<td>249</td>
</tr>
<tr>
<td>Video production, MPEG2, 15-50 Mbps</td>
<td>720x480/576</td>
<td>4:2:2</td>
<td>60I/50I</td>
<td>166</td>
</tr>
<tr>
<td>CCIR601</td>
<td>720x480/576</td>
<td>4:2:0</td>
<td>60I/50I</td>
<td>249</td>
</tr>
<tr>
<td>High quality video distribution (DVD, SDTV), MPEG2, 4-10 Mbps</td>
<td>720x480/576</td>
<td>4:2:0</td>
<td>60I/50I</td>
<td>124</td>
</tr>
<tr>
<td>CCIR601</td>
<td>352x240/288</td>
<td>4:2:0</td>
<td>30P/25P</td>
<td>30</td>
</tr>
<tr>
<td>Intermediate quality video distribution (VCD, WWW), MPEG1, 1.5 Mbps</td>
<td>352x240/288</td>
<td>4:2:0</td>
<td>30P/25P</td>
<td>30</td>
</tr>
<tr>
<td>Video conferencing over ISDN/Internet, H.261/H.263, 128-384 Kbps</td>
<td>352x240/288</td>
<td>4:2:0</td>
<td>30P</td>
<td>37</td>
</tr>
<tr>
<td>Video telephony over wired/wireless modem, H.263, 20-64 Kbps</td>
<td>176x144</td>
<td>4:2:0</td>
<td>30P</td>
<td>9.1</td>
</tr>
</tbody>
</table>

Table 1  Digital Video Formats for Different Applications

Standardization of compression algorithms for video was first initiated by CCITT for teleconferencing and video telephony. A specialist group of CCITT (now named ITU-T, the International Telecommunication Unit – Telecommunication Sector) is chartered with standardizing video compression techniques for video conferencing. The standard (established in
1990) is known as H.320, with H.261 being the video coding part. The standard is developed for video conferencing using the ISDN (integrated service digital network) line, which has a bandwidth $p \times 64$ Kbps, where $p=1,2,\ldots,30$. The video source is either CIF (352x288, 30 fps) or QCIF (176x144, 30 fps), and the video together with the associated audio must be compressed to a multiple of 64 Kbps. Typically, a CIF video requires 384 Kbps or above, and a QCIF video requires 64 Kbps or above, for an acceptable quality. Following the establishment of the H.320 standard, the ITU-T continued the development for other transmission media. The most notable is the H.323 standard, which is targeted for networks with non-guaranteed quality of service (QoS), such as the Internet, and the H.324 standard, which is targeted for very low bandwidth lines, such as plain old telephone lines via 28.8 Kbps modems, or wireless channels. The video coding standard used in both H.323 and H.324 is H.263, which offers significant improvement over H.261, especially at lower bit rate. With H.263, a QCIF video can be compressed down to about 24 Kbps with about the same or better quality than that by H.261 at 64 Kbps.

In parallel with the ITU-T effort, a specialist group, the Motion Picture Expert Group (MPEG), of the ISO (International Standards Organization) also developed various standards for video storage, distribution, and broadcasting. The first standard is known as MPEG-1, which enables the compression of SIF resolution (352x240 pels/sec, 30 fps) video and audio down to 1.5 Mbps with decent quality. This standard enables the storage and playback of movies on CD-ROMs, which at that time have a limited access rate of 1.5 Mbps. The flourish of MPEG1 movies on CDs (Video-CD or VCD) is a key tuning point in digital video development: marking the entering of digital video into the consumer market. When MPEG1 standard was first developed, a dedicated hardware was still required for encoding and decoding in real-time. The appearance of the Intel Pentium I microprocessor soon made it possible to decode MPEG1 movies in real-time, making down-loading MPEG1 movie from the Web a reality. After completion of the MPEG-1 effort, the MPEG focused on standardization of video & audio compression schemes for high quality video broadcasting applications, with a goal of compressing the CCIR601 resolution video (TV quality) down to 3-10 Mbps. This effort leads to the MPEG-2 standard. The establishment of the MPEG-2 standard made possible several important events: video broadcasting from Satellite (direct-TV and the like), the DVD movies, and the digital TV (still emerging). The MPEG-2 standard also includes provisions for compression of HDTV resolution video. It can also handle the MPEG-1 resolution (SIF) and is downward compatible with MPEG-1. Table 2 summarizes the above standards for compression of multimedia signals.

Following the MPEG-2 effort, another standard, MPEG-4, has been developed, which aims to provide accessibility to individual objects in a video scene. The video is coded in an object-based method: i.e. each object is coded separately. An on-going effort of the MPEG group is MPEG-7, which aims to provide a standard description/indexing scheme that can describe the semantic content of the embedded audio and visual information, to enable efficient access/browsing of digital video.

In this experiment, we will introduce the basic techniques for video compression, mainly motion compensated prediction, and briefly review the H.261, MPEG-1 and MPEG-2 coding standards. For a more detailed exposition to this subject, see [1].
II Theories and Schemes for Video Compression

II.1 Motion Estimation & Motion Compensation

Motion estimation and motion compensation is the basis for most video compression algorithms. Motion compensation assumes that the current picture is a slight variation of a previous picture. This creates the opportunity for using prediction and interpolation. When a frame is used as a reference, subsequent frames may differ slightly as a result of movement of the objects or a moving camera, as shown in Fig. 1. To make it easier to compare frames, a frame is not encoded as a whole. Rather, it is split into blocks, and the blocks are encoded independently. For each block in the frame to be encoded (that is, the current frame being addressed), the best matching block in the reference frame is searched among a number of candidate blocks. For each block, a motion vector is generated, which is the difference between the locations of this block and its best matching block in the reference frame. This can be determined by an exhaustive search, a method that we will explain shortly. In the interpolation approach, the motion vectors are generated in relation to two reference frames, one from the past reference frame and another from the next reference frame. The best-matching blocks in both reference frames are searched, and resulting two blocks are averaged.

<table>
<thead>
<tr>
<th>Standards</th>
<th>Application</th>
<th>Video Format</th>
<th>Raw Data Rate (Mbps)</th>
<th>Compressed Data Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.320/H.261</td>
<td>Video conferencing/ telephony over ISDN</td>
<td>CIF/QCIF</td>
<td>37/9.1</td>
<td>&gt;=384 Kbps/ &gt;=64 Kbps</td>
</tr>
<tr>
<td>H.323/H.263</td>
<td>Video conferencing over Internet</td>
<td>4CIF/CIF/QCIF</td>
<td></td>
<td>&gt;=64 Kbps</td>
</tr>
<tr>
<td>H.324/H.263</td>
<td>Video over phone lines/ wireless</td>
<td>QCIF</td>
<td>9.1</td>
<td>&gt;=18 Kbps</td>
</tr>
<tr>
<td>MPEG-1</td>
<td>Video distribution on CD/WWW</td>
<td>CIF</td>
<td>30</td>
<td>1.5</td>
</tr>
<tr>
<td>MPEG-2</td>
<td>Video distribution on DVD/digital TV HDTV</td>
<td>CCIR601 4:2:0</td>
<td>128</td>
<td>3-10</td>
</tr>
</tbody>
</table>

Table 2 Video Coding Standards for Different Applications
II.1.1 The Block Matching Algorithm (BMA) for Motion Estimation

Assume that each block $B_n$ in frame $f_k$ corresponds to a block in $f_{k-1}$ with the displacement $\overline{D}_n$, i.e.

$$f_k(\overline{x} \in B_n) = f_{k-1}(\overline{x} - \overline{D}_n) \quad (10.2)$$

where $\overline{D}_n$ is called the motion vector of $B_n$.

The estimation of $\overline{D}_n$ can be achieved by minimizing a prediction error, such as the sum of squared error or absolute error. These are defined as

Mean Square Error (MSE):

$$PE(\overline{D}_n) = \sum_{\overline{x} \in B_n} [f_k(\overline{x}) - f_{k-1}(\overline{x} - \overline{D}_n)]^2 \quad (10.3)$$

and

Mean Absolute Difference (MAD):

$$PE(\overline{D}_n) = \sum_{\overline{x} \in B_n} |f_k(\overline{x}) - f_{k-1}(\overline{x} - \overline{D}_n)| \quad (10.4)$$

The BMA uses an exhaustive search procedure to find the block that has the least error. More specifically, it compares the present block with every possible block in a predefined range surrounding the current position. For each candidate block, it calculates the prediction error. After
all possible candidates have been investigated, the one with the least error is considered the best matching block.

The actual motion between two video frames is usually not block-wise constant. Therefore, using block matching algorithm for motion estimation and compensation cannot produce accurate prediction always. When the actual motion in a block is not a simple translation, the algorithm tries to find the block that produces the least matching error. Figure 2 shows an example of motion compensation using block matching algorithms. The top-left is a previous frame, the top right is a current frame, the bottom right is the estimated motion field (i.e. we draw the estimated motion vector at each block center), and the bottom left is the predicted image using the estimated motion field. We can see that the algorithm accurately determines the zero motion in the background and the overall horizontal translation in the head. But the eye closing was not produced by the simple block-wise motion model. There are blocking artifacts at the boundary between the foreground object and the background.

![Figure 2 Example of Motion Compensated Prediction Using Block Matching Algorithm](image)
II.1.2 Video Coding Using Motion Compensation and Transform Coding

The most popular video coding method is known as block-based hybrid coding. It divides each frame into macroblocks (MBs), each containing several 8x8 blocks. Each MB is coded using a combination of motion compensated temporal prediction and transform coding using DCT, as shown in Figure 3. Each MB can be coded in one of the two modes. In the intra-mode, a DCT-based coding method (similar to JPEG) is used directly on each block. This mode is used for every MB in the first frame, and then periodically inserted in subsequent frames. In the inter-mode, a motion vector is first determined, and the DCT method is used to code the motion compensation error. Specifically, it predicts the current MB (luminance component only) of the current frame with the best matching MB in the previous frame. If the prediction error is below a predefined threshold, no differential data are specified. Otherwise, the prediction error is transformed using the DCT, and the transformed coefficients are quantized and encoded using a runlength coding method similar to the JPEG standard. Finally, the encoded bit stream is sent to the video multiplex along with the coded motion vector information. As described in the JPEG description, the quantizer’s step sizes can be adjusted based on desired picture quality and coding efficiency.

II.2 H.261 Video Coding Algorithm (Px64)

In 1990, the CCITT approved a set of international video conferencing standards, including the H.261 Video Codec, for Audiovisual Services over ISDN at px64 Kbps (also known as the px64 standard). The intended applications for this standard are videophone and video conferencing systems. Consequently, systems incorporating this standard must be able to code and decode this standard in real time. The value of p ranges from 1 to 30. For a basic rate ISDN
The CCITT adopted CIF (Common Intermediate Format) and QCIF (Quarter-CIF) as the video formats for visual telephony. The CIF format video has 352x288 pels for Y and 176x144 pels for Cr & Cb frames. The QCIF video is a quarter size of CIF, with 176x144 pels for Y, 88x72 pels for Cr & Cb. The frame rate can be varied between 1 to 30 frames/sec. All codecs are required to operate at the QCIF level; operation at the CIF level is optional. At approximately 30 frames/sec (29.97 is the maximum supported), uncompressed CIF includes 36.45 Mbits/sec of information and QCIF at 9.115 Mbits/sec. Significant reduction is required for a 1.5 Mbits/sec channel but as much as 24:1 is needed for a p=2 channel (128 Kbits/sec). Typically, CIF is recommended for transmission exceeding 384 Kbits/sec(p=6).

The encoder for H.261 uses a hybrid of DCT and DPCM schemes with motion estimation, as shown in Fig.3 Each MB can be coded in one of the two modes: Intra-mode or Intermode. The intra-mode is used for every MB in the first frame, and then periodically inserted in subsequent frames, to prevent the propagation of errors caused by transmission errors indefinitely.

The H.261 use a hierarchical data structure for encoding data. These include Picture, Group of Blocks (GOB), Macro Block (MB) and block. A block is an 8x8 pixel block, which could contain either Y, Cb or Cr samples. A MB is composed of four 8x8 luminance blocks (Y) and two chrominance blocks (Cr & Cb). A GOB consists of MBs in several consecutive rows. A picture consists of several GOBs. The H.261 standard defines the syntax of the coded bit stream. Each block contains the DCT coefficients (TCOEFF) of a block followed by an EOB marker. Each MB consists of data for 6 blocks and an MB header. A GOB consists of a GOB header followed by the MBs in the GOB. Finally, the picture layer consists of a Picture header enveloped around a sequential array of GOBs.

II.3 The MPEG-1 Standard

The MPEG-1 standard is developed primarily for storage of full motion video at SIF resolution (352x240 pels for Y, 176x120 pels for Cr & Cb at 30 fps) at 1.5 Mbps. In addition to the requirement of full-motion rendering at 1.5 Mbps, the compressed bit stream must allow random access such as fast forward and rewind. In the following, we discuss how these requirements are addressed by MPEG1.

II.3.1 Picture Coding Modes and Bi-Directional Motion Compensation

One major difference between MPEG-1 and H.261 is that, in addition to using motion compensated from the previous frame, it also uses prediction from a future (but previously coded) frame. More generally, a current frame can be predicted from both a previous and a future reference frame. This is called bi-directional prediction in MPEG. MPEG-1 codes the frames in a video in 3 possible modes: I-picture, P-picture, and B-picture. The frames are divided into group of pictures (GOPs) so that each GOP consists of one I-picture, and several P-pictures and several B-pictures. This is illustrated in Fig. 4. The operations for different pictures are explained in more detail below.
Intrapictures (I): An I-picture is compressed based on this picture alone, using a method very similar to JPEG. As in the JPEG algorithm, each 8x8 block in a 16x16 MB is fed through the DCT module to generate DCT coefficients. The coefficients are then quantized and organized in zig-zag order to give the best run length of zero coefficients. The runlength & nonzero values are then coded using the Huffman coding method. The I-picture mode is used in the first frame in each GOP, mainly to enable random access.

Unidirectional Predicted pictures (P): The P-picture mode uses motion compensated prediction technique for compression much like the H.261 standard. Each MB uses the previous P picture or I picture (anchor pictures) to estimate a motion vector. The prediction error is then transformed using the DCT. The DCT coefficients are then quantized and runlength coded.

Bidirectionally predicted pictures (B): The B-picture mode utilizes one of three types of motion compensation techniques for each MB: forward motion compensation, backward motion compensation and interpolative compensation. Forward motion compensation uses the past anchor picture (I or P) information as in the P-picture. Backward motion compensation uses future anchor picture information by which the current MB is predicted by a best matching block in a future P- or I-picture. Interpolative compensation uses the average of the best matching blocks in both the past and future anchor pictures. Bidirectional motion compensated prediction is illustrated in Fig.5.

Because all the frames in a GOP can be decoded without the knowledge of the previous GOP, a GOP is the basic unit for random access. A fast forward can be realized by decoding I-picture only or I- and P-picture only. A fast rewind can be realized by decoding only I-pictures.
II.3.2 Half-Pel Accuracy Motion Estimation

Another difference between the MPEG-1 motion estimation and H.261 is that the motion vectors in MPEG-1 are estimated to half-pel accuracy. That is, the exhaustive search is done with a half-pel increment, rather than on integer pel increment. The half-pel motion estimation is more accurate, enabling the reduction of prediction error. But it also requires the interpolation of the previous frame to produce samples at half-pel locations, which requires more computation than integer-pel accuracy motion estimation.

II.4 The MPEG-2 Video Coding Standard

As mentioned in the introduction, the MPEG-2 standard is mainly developed for coding CCIR601 video. The basic coding method in MPEG-2 is the same as that in MPEG-1, with the same GOP structure and each MB is coded using either DCT directly (I-mode), with unidirectional prediction (P-mode), or with bi-directional prediction (B mode). Apart from the higher spatial resolution, the main difference between CCIR601 and CIF/SIF video is the use of interlacing in CCIR601 video. This significantly complicates the compression process. Therefore, special methods have been developed to handle interlaced pictures, which changes the way motion estimation/compensation and DCT is performed. A more detailed discussion can be found in [1].

MPEG-2 can handle multiple video formats with different resolution levels. It also has different profiles that offer increasingly more functionalities. The above discussion only refers to the main profile at main level (written as mp@ml). A HDTV signal is coded using the main profile at high level. Another important development of MPEG-2 beyond that of MPEG-1 is the scalability profile, which enables a video be coded into a base-layer and an enhancement layer. The base layer provides a base-line quality, and the enhancement layer, when added to the base layer, can further improve the quality. An MPEG-2 coded video using a scalability mode can be transported over networks with varying bandwidth or to receivers capable of receiving different spatial resolutions. Figure 6 summarizes the various profiles and levels supported by MPEG-2.
II.5 Other Commercial Video Compression Standards

II.5.1 Intel’s Indeo Technology

Developed by the Intel Architecture Labs, Indeo video is a software technology that reduces the size of uncompressed digital video files from five to ten times. The Indeo technology has been bundled in products like Microsoft’s Video for Windows and Apple’s QuickTime.

Indeo technology uses multiple type of “lossy” and “lossless” compression techniques. Indeo technology compresses video in real time as it is being recorded through a video capture board; thus, the uncompressed data does not have to be saved on disk. Analog video received from a video camera, VCR, or laser disk in any standard format such as NTSC is converted into digital format by video capture board such as an Intel Smart Video Recorder board.

The Indeo method includes the following steps (not all are required):

1. YUV sampling, to reduce the pixel area to an average color value
2. Pixel differencing and temporal compression, to shrink data by storing only the information that changes between pixels or frames (This is accomplished by quantizing the frame difference using vector quantization.)
3. Run-length encoding, to compress the code word indices
4. Variable-content encoding, to reduce a variable amount of information into a fixed number of bits