REIHE INFORMATIK
7/98

Efficient Video Transport over Lossy Networks

Christoph Kuhmünch and Gerald Kühne
Universität Mannheim
Praktische Informatik IV
L15, 16
D-68131 Mannheim

Efficient Video Transport over Lossy Networks

Christoph Kuhmünch and Gerald Kühne
Lehrstuhl für Praktische Informatik IV
University of Mannheim
L 15,16, 68131 Mannheim, Germany
{cjk, kuehne}@pi4.informatik.uni-mannheim.de

Abstract

Nowadays, packet video is an important application of the Internet. Unfortunately the capacity of the Internet is still very heterogeneous because it connects high bandwidth ATM networks as well as low bandwidth ISDN dial in lines. The MPEG-2 and MPEG-4 video compression standards provide efficient video encoding for high and low bandwidth media streams. In particular they include two paradigms which make those standards suitable for the transmission of video via heterogeneous networks. Both support *layered video* streams and MPEG-4 additionally allows the independent coding of *video objects*. In this paper we discuss those two paradigms, give an overview of the MPEG video compression standards and describe transport protocols for Real Time Media transport over lossy networks. Furthermore, we propose a real-time segmentation approach for extracting video objects in teleteaching scenarios.

Keywords: *video compression standards, hierarchical encoding, real time transmission*

1 Introduction

Video transmission became an important application of the Internet. With the development of the MBone [Dee89, Dee91] and its tools [McC95] multipoint transmission of media streams became possible. Typical scenarios are for example Multipoint Videoconferencing, Video on Demand Services or Teleteaching [Eck97]. Unfortunately the capacity of the Internet is still very heterogeneous because it connects high bandwidth ATM networks as well as low bandwidth ISDN dial in lines. Common video compression standards already allow the adaption of the compression rate and thereby of the video quality according to the network connection of a *single* receiver side. However they fail if many receivers with different network capacities receive the stream. Consider the following example as presented in Figure 1: A teleteaching lecture has to be transmitted from classroom *A* to the remote classrooms *B* and *C* as well as to a student at home via ISDN. Classroom *A* and *B* share a high speed local area network while classroom *C* is connected via a 2MBit/s ATM connection. If common encoding schemes are used, all receivers have to accept the quality of the receiver with the lowest bandwidth of 128kbit/s in our case. Otherwise the “weaker” receivers will suffer from high packet loss.

On the other hand, the MPEG video compression standards provide efficient video encoding for high and low bandwidth video streams. In particular MPEG-2 and MPEG-4 implement two paradigms which make those standards suitable for the transmission of video via heterogeneous internetworks. Both support *layered video* streams. Additionally MPEG-4 allows the independent coding and transmission of *video objects*.

The remainder of the paper is structured as follows: Section 2 summarizes MPEG compression fundamentals followed by an overview of hierarchical encoding schemes. Section 4 surveys the implementation of compression and scaling methods in MPEG video encoding standards concentrating on MPEG-2 and MPEG-4. Afterwards the Real Time Transport Protocol RTP is described. This protocol provides transport services for transmitting video streams over datagram networks. In Section 6 we describe the integration of MPEG-2 video into the videoconferencing tool vic and propose a real-time segmentation approach for extracting video

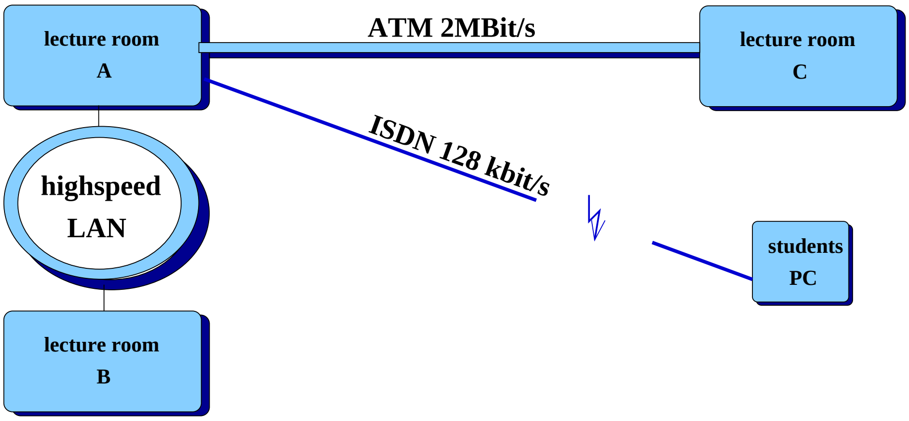

Figure 1: Example: Transmission of a Teleteaching Lecture to receivers with different network capacities.

objects in the context of teleteaching scenarios. Finally, Section 7 offers outlook and concluding comments.

2 MPEG video compression fundamentals

Consider a color image in approximate TV size of 720×576 picture elements (pixels). If we assume the use of the RGB color model, we need 3 bytes for each pixel — one for each color component in the interval $[0, 255]$ —, i.e. $720 \times 576 \times 3 \approx 1$ Mbyte when storing the complete image in computer memory. Consider further a video sequence with a frame rate of 25 color images per second. To distribute such a video stream in real time, we have to transmit about 200 Mbit per second. However, even with advanced network technologies and increasing bandwidth this remains a problem. Therefore efficient compression techniques are a prerequisite for transmission and distribution of image and video data over computer networks.

In the following we will concentrate on the key techniques of the MPEG video compression standards. The efficiency of MPEG video compression relies on three principles: (1) Subsampling of pictures, (2) spatial redundancy reduction and (3) temporal redundancy reduction. These principles exploit the significant amount of statistical and subjective redundancy contained within and between consecutive pictures of a video sequence.

2.1 Subsampling

The “quality” needed in an image depends on the sensitivity of the human visual system to changes in intensity. Human vision has a poor response to changes in chromaticity compared to its response to changes in brightness [Poy96, Pen93].

Therefore, while full brightness information should be maintained, the data capacity accorded to the color information in a picture can be reduced using subsampling techniques. The general concept behind subsampling is to reduce the size of the input data. Considering a sequence of images, this can be done by subsampling in the spatial dimension of the single picture or in the temporal dimension of the entire sequence. A simple technique to reduce an image in horizontal and vertical dimension (spatial subsampling) is to skip every second pixel along both directions. Consequently, this procedure applied to a picture of size 256×256 results in a 128×128 image.

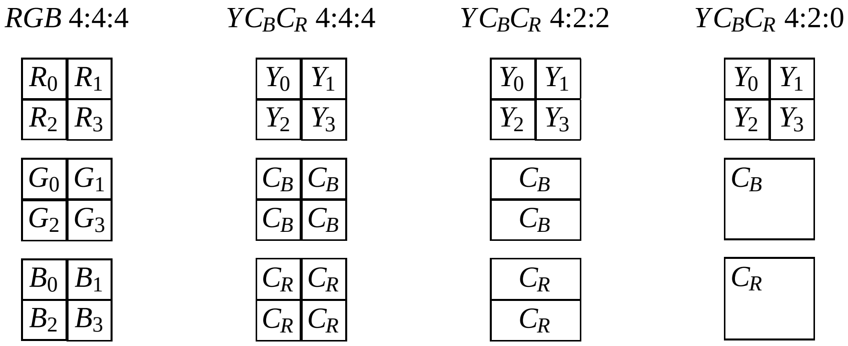

Figure 2: Subsampling of chrominance components. A 2×2 *matrix of RGB pixels is transformed to* YC_BC_R*. The chrominance components* C_B *and* C_R *are subsampled with the ratios 4:2:2 and 4:2:0, respectively [Poy96].*

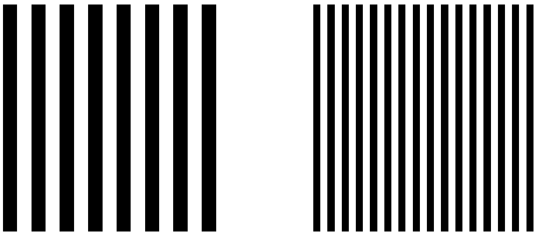

Figure 3: Influence of spatial frequency on human perception of contrast. The gratings differ in there spatial frequency. [Gol93].

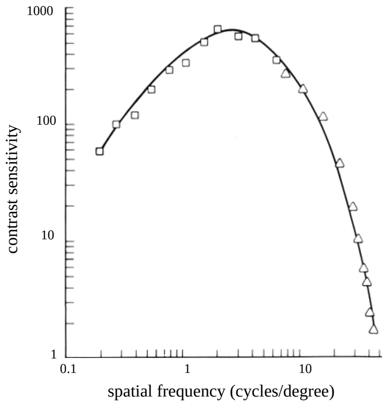

Figure 4: A contrast sensitivity function (CSF) for a sine-wave grating [Gol93].

In the MPEG standards, images are divided into three rectangular matrices of integers: a luminance matrix (Y) which represents brightness and two chrominance matrices (C_B, C_R) which represent the color information [ISO93, ISO95, ISO97].

To exploit the characteristics of human vision mentioned above, the chrominance components are subsampled relative to the luminance component (see Figure 2). For example, when applying the ratio 4:2:0 to the three components, the C_B and the C_R components are one half the size of the Y matrix in both horizontal and vertical dimension.

2.2 Spatial redundancy reduction

In Figure 3 two gratings of alternating black and white bars are shown. These images differ in the number of repeating patterns per unit distance, where one repeating pattern or cycle consists of one black and one white bar. The number of cycles per unit distance is called spatial frequency. Consequently, the grating on the right has a higher spatial frequency than the one on the left.

When viewed from a distance of about one meter, the high-spatial-frequency pattern on the right appears to have less contrast than the grating on the left. How can we explain this perception? When increasing the distance between observer and the patterns, the retinal image becomes smaller or in other words the visual angle decreases. If we define the spatial frequency as number of cycles per degree of visual angle, we determine that the spatial frequencies of the gratings increase. Hence, our perception of contrast is affected by the spatial frequency of the stimulus [Gol93].

The plot of the physical contrast needed to see a grating versus the grating's spatial frequency — the contrast sensitivity function (CSF) — depicts this relation (see Figure 4). We can gather from the CSF that (1) the visual system is most sensitive to gratings with spatial frequencies of about three cycles per degree of visual angle and (2) that the sensitivity decreases from this point until a certain threshold is reached where human perception can not detect repeating patterns anymore.

Therefore, if we calculate spatial frequencies from an image captured from a scene in our environment, we can omit imperceptible frequencies and encode those frequencies the visual system is most sensitive to with high accuracy.

To obtain spatial frequencies from a given digital image, an image transform is needed. In most practical case, transform schemes based on the two-dimensional discrete cosine transform (DCT) provide best image quality for a given bit rate.

The DCT decomposes the image to a weighted sum of cosine basis functions. These functions are comparable to the gratings mentioned above and differ in their spatial frequencies. The $N \times M$ matrix of pixels is transformed to a $N \times M$ matrix of coefficients for the different basis functions. The coefficient corresponding to the basis function with the lowest spatial frequency is called DC coefficient. The other coefficients are called AC coefficients. On account of the properties of the human visual system discussed above, a visually weighted quantization can be applied to the coefficients. Each coefficient is divided by an appropriate quantization factor[1]. For example, coefficients corresponding to high spatial frequencies are divided by large values, because coarse granularity is sufficient to suit the contrast sensitivity of the visual system in this area.

The quantization process tends to make many coefficients zero. Therefore, the matrix of quantized coefficients can be efficiently encoded by entropy coding techniques [Gol93].

2.3 Temporal redundancy reduction

If we compare two consecutive frames of a video sequence, we will find only slight or even no changes in the position of the objects contained in the sequence. That means a video sequence contains much temporal redundancy, which can be reduced by temporal motion compensation. Motion compensation is based on the estimation of motion between video frames. Ideally, for each pixel a motion vector can be determined that provides an offset from the coordinate position in the current picture to the coordinates in the previous picture. However, encoding one motion vector for each pixel would result in an enormous overhead. Since it is likely that adjacent pixels belong to the same object and move therefore in the same direction, it is useful to encode motion vectors on a per image block basis (e.g. 16×16 pixels).

However, for the reconstruction of a picture from previous frames by means of motion estimation, the sole encoding of block based motion vectors is only sufficient in the case where we have blocky objects which move in parallel to the image plane. In natural video this is obviously not the case. Consequently, additional information is needed for the reconstruction process. Therefore, the prediction error—the difference between original image and motion compensated image—is encoded. To reduce spatial redundancies in these prediction error images, they are encoded using the discrete cosine transform.

[1] Each coefficient is calculated by measuring the threshold for visibility of the corresponding basis function.

3 Hierarchical encoding and layered video

Common video encoding and compression techniques as described in Section 2 already allow the adaption of the compression rate and thereby of the quality of the video to the available bandwidth. As the example in Figure 1 – described in Section 1 – proves these techniques fail if a video is transmitted simultaneously to several receivers with different network capacities. In order to solve this problem *hierarchical* or *layered encoding* schemes have been developed. The idea of these schemes is to encode video signals not only into one but into several output streams. Each stream S_i depends on all lower streams $S_0, \ldots, S_{i-1}$, in other words it can only be decoded together with these lower streams. Each stream adds to the quality of the transmitted video. For example the lowest stream S_0 only provides a quality scaled down to a low resolution black and white video with a few frames per second. With each addition of a higher stream the quality gradually rises to high resolution color video with 25 frames per second. Each receiver side can adjust its quality according to the available bandwidth by simply joining the appropriate number of streams. Figure 5 summarizes this method.

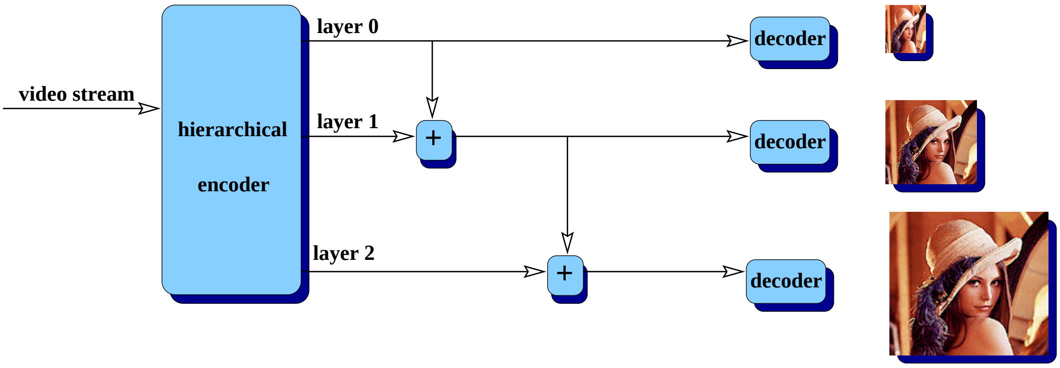

Figure 5: The Encoder produces a set of depending streams and transmits each stream individually. The decoder combines the streams into a single video. With each added stream the quality gradually progresses ([McC96]).

Conclusively, hierarchical encoding techniques scale the video quality in the three dimensions color resolution, spatial resolution and temporal resolution which are described in more detail in the following paragraphs.

3.1 Three dimensions of video

We define the following three dimensions of video:

- number of bits per pixel (color resolution)
- number of pixels per frame (spatial dimension)
- number of frames per second (temporal dimension)

3.1.1 Color resolution

A digital image consists of discrete pixels in each row and column. Typically three bytes (24 bits) are used to represent the color value of a pixel. There exist several color models but for video compression a pixel is usually represented by the triple (Y, C_B, C_R) where Y represents the luminance, C_B the chrominance and C_R the color hue of the pixel (see Section 2.1).

3.1.2 Spatial dimension

The spatial dimension describes the horizontal and the vertical resolution of each picture the video consists of. The ITU defines a common intermediate format (CIF) with a size of 352×288 pixels. The picture area covered by this format has an aspect ratio of $4:3$ and corresponds to the active portion of the standard video input [ITU90]. The picture size of a TV signal after digitalization of the European TV standard PAL is approximately 720×576 pixels. The American TV standard NTSC defines the picture size with about 640×480 pixels [Tek95].

3.1.3 Temporal dimension

The temporal dimension describes the temporal resolution of the video, in other words the number of frames per second. The number of frames per second in European TV is 25 frames per second the American standard is 30 frames per second [Poy96].

In the following we use these three dimensions in order to define *hierarchical encoding* more precisely.

3.2 Definition of hierarchical encoding

Before we describe several approaches in more detail we give a precise definition for *hierarchical encoding*. The following definition is a generalization of the one Steven McCanne describes in his PhD thesis [McC96].

Definition.

A color video V consists of a sequence of frames $V = \{F_1, F_2, ...\} \mid F_i \in [0,255]^{w \times h \times 3}$ (temporal dimension) where each frame consists of $w \times h$ pixel (spatial dimension) and each pixel (color resolution) typically is represented by a triple $(Y, C_B, C_R) \in [0,255]^3$ for the luminance, chrominance and color hue of the pixel (see Section 3.1.1). Let $V_{i,k}$ be a subsequence of V with the length k starting at frame i:

$$V_{i,k} = \{F_i, ..., F_{i+k}\} \tag{1}$$

Common video compression techniques encode image sequences by reducing the spatial and temporal redundancy which means that they encode a subsequence $V_{i,k}$ of frames into a *single* code $C_{i,k}$. On the other hand a hierarchical encoder E encodes a sequence of k frames into *several* output codes $C^1, ..., C^L$. Therefore we define E in the following way:

$$E : V_{i,k} \rightarrow \{C^1_{i,k}, ..., C^L_{i,k}\} \tag{2}$$

In order to reassemble the video at the receiver side we need a Decoder D which combines the codes C back into a sequence of frames $\{\hat{F}_1, ..., \hat{F}_k\}$.

$$D : \{C^1_{i,k}, ..., C^l_{i,k}\} \rightarrow \hat{V}_{i,k} \quad \mid \quad l \leq L \tag{3}$$

Note, that the difference between the original subsequence $V_{i,k}$ and the reassembled sequence $\hat{V}_{i,k}$ is getting smaller the more codes are taken into account.

According to this definition the elementary task of a hierarchical encoder E is to define encoding schemes which split (and compress) a given frame sequence into a set of codes C. Consequently in the next Section we explain several encoding schemes.

3.3 Layered compression algorithms

In the recent past a number of hierarchical video compression techniques have been developed which scale and compress a frame sequence in its three dimensions time, size and color depth. In the following we concentrate on the spatial and the temporal dimension.

3.3.1 Spatial compression and scaling

Many hierarchical compression schemes concentrate on the spatial dimension. Hence, each subsequence $V_{i,k}$ defined in Equation 1 has the length of $k = 1$ frame and each code C^l encodes one frame. Layered compression standards which scale video in its spatial dimension can be split into two categories. Algorithms of the first category can be summarized with *layered DCT* and the second with *pyramid coding*.

Layered DCT

Techniques which fall in this category are based on the compression technique described in Section 2.2. There are two different approaches to split the video stream into different layers.

- layered frequencies
- layered quantization

In the *layered frequencies* approach each 8×8 block of each image of the video is transformed into the frequency domain using DCT. Afterwards the DC coefficient and the AC coefficients are quantized but the quantized coefficients are not entropy encoded in a single step, rather they are grouped in subsets and each subset l is entropy encoded in a code C^l.

In the *layered quantization* approach each block of each image is also transformed into the frequency domain. Afterwards the DC coefficient is quantized in a single step and the resulting value is encoded in the base layer C^0. But instead of quantizing the AC coefficients in a single step the precision of the value is gradually progressed. For example in the “Progressive JPEG standard” defined by the Independent JPEG Group each AC coefficient is refined one bit at a time. In other words, each code C^l contains bit plane number l of the AC coefficients. Although this technique is widely used in the world wide web in order to gradually display images on web pages, this approach suffers from a drawback which limits its usefulness for real time video transmission. The successive reconstruction of an image one bit plane at each time is very time consuming. Therefore Amir et al. [Ami96] defined a more general approach called *LDCT*. They encode AC coefficients with 9 Bit accuracy and split them into four groups. The first group contains the three most significant bits of each coefficient, while the remaining three groups encode two bits each. Each of the four groups is transmitted on a different layer. At the decoder side the AC coefficients are reassembled from the four different layers. If the decoder receives only the base layer a coarser image can still be reconstructed.

Laplacian Pyramid Coding

Similar to the former algorithms this approach also processes each single frame of the video separately. The core idea of this approach [Bur83] is described in Figure 6. The encoder first downscales the image, compresses it according to the DCT-based encoding technique and then transmits it in the base layer stream. When the image is decompressed and upsampled again a much coarser copy arises. To compensate the difference the decoder subtracts the resulting copy from the original image and sends the encoded differential picture in the enhancement layer stream.

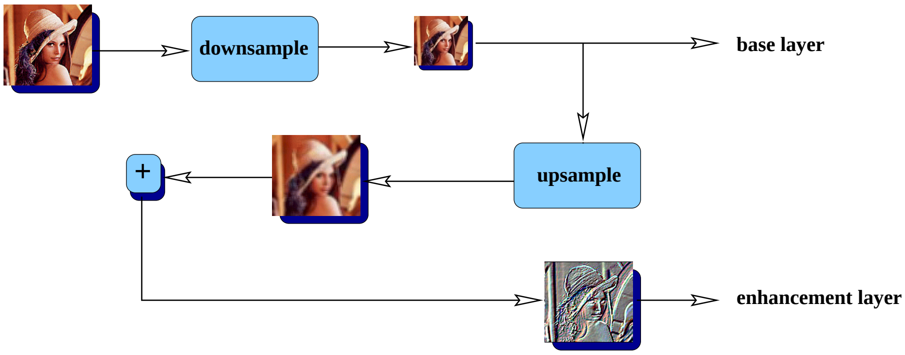

Figure 6: Data flow of a Laplacian Encoder ([McC96]

This method results in a two layer hierarchy where the base layer contains a coarser version of the original image and the enhancement layer the difference between the original image and the coarser one. In order to gain more than just two layers the algorithm can be recursively extended in the following way: In n recursion steps the original image I_0 is scaled down (e.g. by half at each step) resulting in a very coarse image I_n. This very coarse image I_n is encoded and transmitted in the lowest layer C^0. At the lowest recursion level n image I_n is upscaled and subtracted from I_{n-1}. The resulting differential picture is encoded and transmitted at enhancement layer C^1. In the next higher recursion level I_{n-1} is subtracted from I_{n-2} and so on.

3.3.2 Temporal compression and scaling

At first glance temporal scaling seems to be an easily implementable way to scale video. Indeed temporal scaling can be accomplished without much effort if each frame is compressed independently without motion compensation. In this case the frames can be freely spread over different layers. Figure 7 describes a possible approach with three layers where a subsample of the image sequence is transmitted on each layer. Merz et. al. [Mer97] describe a Web-Movie System which makes use of this approach.

Difficulties arise if the compression scheme takes motion compensation into account. Consider an MPEG compressed image sequence where dependencies between subsequent encoded frames exist due to motion compensation. If we distribute those frames on several layers as described above without taking this dependency into account decoders which receive only some layers will not be able to decode all the frames they receive. There are two possible ways to scale motion compensated video. The first approach is to encode the video independently on each layer. This approach results in transmitting groups of pictures (GOPs) on each layer. Between such groups dependencies are avoided. The second approach takes the structure of the GOPs into account. In Figure 8 a possible scaling method with three layers is described. All independent coded frames (I-frames) have to be transmitted on the base layer. On the second layer the predictive-code frames (P-frames) are transmitted and the highest layer transports the bidirectinally predictive-coded frames (B-frames) (see Section 4 for further explanation of the different frame types).

3.3.3 Hybrid scaling

It is difficult to decide if temporal or spatial scaling is better suited for a certain application scenario. For example for the transmission of an action film it might be better to send the full frame rate with reduced image quality but for a teleteaching scenario were the teacher writes

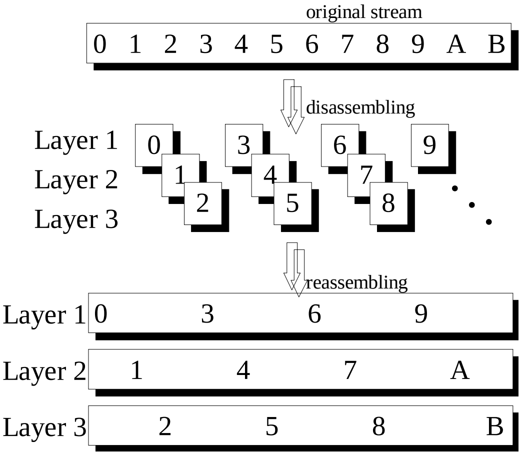

Figure 7: Temporal scaling of a video stream where each frame is encoded independently without motion compensation

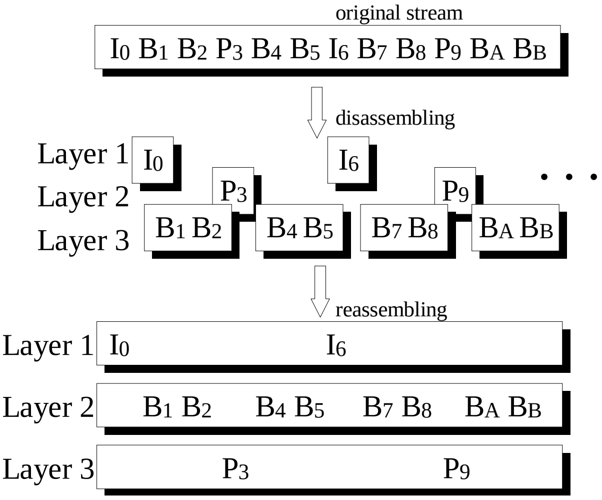

Figure 8: Temporal scaling of a video stream compressed with motion compensation

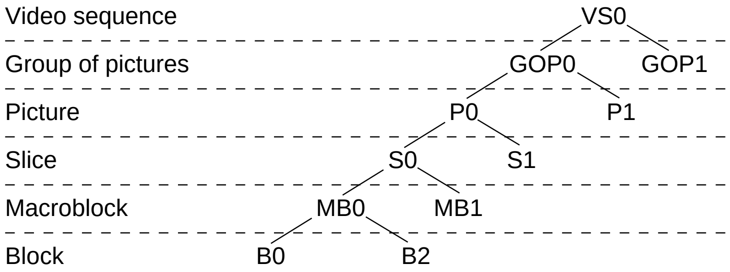

Figure 9: Syntactic structure of an MPEG-2 Video encoded data stream

on the blackboard it might be more important to receive a lower frame rate with high quality images.

The most flexible way to scale video is using a hybrid approach which scales in both—temporal and spatial—dimension. That way the video quality can be gracefully adapted to the demands of the applications and the perception of the user. There are two possible ways to accomplish this:

(1) The first approach starts with temporal scaling, which results in a set of layers containing a subset of the images of the video sequence. The images on the different layers now are spatially scaled into a set of sublayers for each temporal layer.

(2) Alternatively it is possible to perform the spatial segmentation first. This results in a set of layers where each layer adds to the quality of the images. The subsamples of the images on each layer are now distributed on sublayeres. Merz et. al. describe both approaches in [Mer97] and found out that the first alternative can be implemented more efficiently.

4 MPEG video compression standards

The standards for coding audio and visual data developed by the Motion Pictures Expert Group (MPEG), namely ISO/IEC 11172 (MPEG-1), ISO/IEC 13818 (MPEG-2) and ISO/IEC 14496 (MPEG-4)[2], has received considerable attention.

In the following we will summarize based on the previous Sections compression techniques and hierarchical encoding schemes provided by the video compression parts of MPEG-2 (ISO/IEC 13818-2) and MPEG-4 (ISO/IEC 14496-2) [ISO93, ISO95, ISO97].

Additionally, error resilience strategies will be discussed. In the context of error prone environments and communication channels those strategies are of special importance. Both MPEG-2 and MPEG-4 support error resilience modes relevant to packet loss and bit errors in transmissions.

4.1 MPEG-2 Video

4.1.1 Structure of coded video data

An MPEG-2 compressed video stream can be thought of as a tree like structure (see Figure 9). We will traverse the tree structure from top to bottom to explain the different hierarchy levels.

[2] The specification of MPEG-4 reached the state of a final committee draft and is expected to become an international standard in early 1999.

The video sequence forms the highest syntactic level and is subdivided into group of pictures, which are intended to assist random access into the sequence.[3]

A picture—the equivalent of a single movie frame—represents the basic unit of display. It is divided into several non-overlapping slices which provide additional information for the decoding process. This structure allows a decoder to recover after a data error and to resynchronize its decoding (see Section 4.1.4). A slice consists of an arbitrary number of macroblocks. A macroblock of size 16×16 pixels contains a Section of the luminance component and the spatially corresponding chrominance components. In MPEG-2 three chroma formats for macroblocks are supported, namely, 4:2:0, 4:2:2 and 4:4:4. According to Section 2.1, a 4:2:0 macroblock consists of 6 blocks of size 8×8 pixels (4 Y, 1 C_B, 1 C_R blocks), a 4:2:2 macroblock contains 8 blocks and a 4:4:4 macroblock consists of 12 blocks.

The arrangement of I-,P- and B-frames can be freely chosen by the user to suit the needs of specific applications. For example, if fast random access is necessary, the user would choose an encoding pattern consisting only of I-frames (*IIIII* ...) to avoid dependencies between frames. However, if compression efficiency is most important, an appropriate encoding pattern would be *IBBPBBI* ....

4.1.2 Compression

There are three types of pictures that use different coding methods:

(1) An Intra-coded (I) picture is encoded independently from any other picture using information only from itself.

(2) A Predictive-coded (P) picture refers to a reference frame. It is coded using motion compensated prediction from a past I- or P-picture.

(3) A Bidirectionally predictive-coded (B) picture is encoded using motion compensated prediction from a past and/or future I- or P-picture.

The first frame in a video sequence is encoded as I-picture using information only from itself. The input frame is partitioned into macroblocks. Then the discrete cosine transform is applied to each 8×8 block (see Section 2.2) and the result is encoded using entropy coding techniques.

Consecutive frames are intra-frame coded (P-/B-frames) using motion compensation techniques described in Section 2.3. For each macroblock of size 16×16 one motion vector and the prediction error is calculated. Additionally, the prediction error macroblock is encoded with the DCT-based technique described above.

4.1.3 Scalability

Three basic scalable coding schemes are provided by MPEG-2. These schemes correspond to the basic hierarchical coding techniques presented in Sections 3.3 and 3.3.2. Each of the basic coding schemes produces a base layer and an enhancement layer.

SNR (quality) scalability relies on the layered DCT approach. *Spatial scalability* employs spatial pyramid coding and in the *temporal scalability* scheme frames are distributed among base and enhancement layer.

It is also possible to form a hybrid scheme from different basic scalability techniques. Hybrid scalable coding schemes involve three layers, one base layer and two enhancement

[3] Note that in MPEG-2 the formation of pictures to group of pictures is optional. It is also possible to encode pictures directly without grouping them.

layers. In the following we give an example for a combination of spatial and SNR scalability. The base layer encodes standard TV resolution at basic quality. Applying SNR scalability, enhancement layer 1 generates standard TV resolution at higher quality from the base layer. Enhancement layer 2, when applied to the combination of base layer and enhancement layer 1, provides HDTV resolution which is coded with spatial scalability.

4.1.4 Error resilience

Error resilience techniques supported by MPEG-2 are summarized in the categories resynchronization and error concealment. The latter covers methods of disguising an error once it has occurred. Techniques from the first category restrict the influence of an error in spatial and/or temporal dimension.

MPEG-2 supports resynchronization in both spatial and temporal dimension. The first is achieved by grouping macroblocks into slices. At the start of a new slice, information called a slice header is placed within the bitstream. The slice header provides information which allows the decoding process to be restarted. The influence of an error in temporal dimension is restricted by intra-coding techniques. An I frame is independent from previous and consecutive frames and therefore limits the propagation of errors through the video sequence. By choosing a picture coding pattern which encodes only I frames, temporal dependencies are eliminated. However, there is a trade off between compression efficiency and the avoidance of temporal dependencies.

If the error can be localized effectively by means of resynchronization, error concealment can be achieved by estimating the lost data from spatial or temporal adjacent data. For example the decoder could replace a lost macroblock with the macroblock in the same location in the previous picture. Spatial predictive concealment could be achieved by compensating a lost macroblock by an interpolation from neighboring macroblocks. However, those approaches are only successful under the assumption that adjacent blocks and pictures possess fairly similar characteristics.

The use of scalable coding techniques mentioned in the previous Section facilitates the concealment process. Packet loss or bit errors in the enhancement layer can be easily compensated by interpolation from the base layer. However, this assumes that the base layer is error protected or transmitted over a reliable connection .

4.2 MPEG-4 Visual

As described in the preceding Section, MPEG-2 video compresses rectangular pictures by exploiting temporal and spatial redundancies. While MPEG-4 derived the basic video compression techniques from MPEG-2, its scope is much wider, as expressed by the term “visual” in the title of [ISO97]. MPEG-4 visual allows the coding of both natural and synthetic video and provides content based access to individual video objects in the scene. Further, it provides enhanced error resilience and scalability.

4.2.1 Structure of coded visual data

According to MPEG-2 video an MPEG-4 encoded data stream can be described by a hierarchy (see Figure 11). The highest syntactic structure is the visual object sequence. It consists of one or more visual objects. Each visual object belongs to one of the following object types: Video object, still texture object, mesh object, face object (see Figure 10). In the context of natural video coding and transmission, we will only cover video objects in detail.

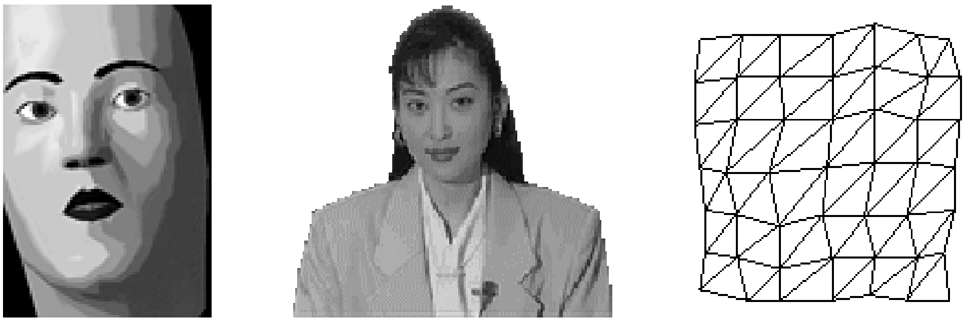

Figure 10: Visual objects. From left to right: synthetic face object, natural video object, 2D mesh object which can be combined with still texture images[Doe96].

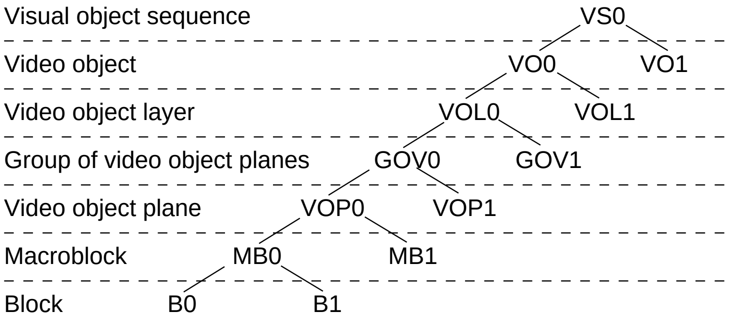

Figure 11: Syntactic structure of an MPEG-4 Visual encoded data stream. Only the parts concerning video objects are shown.

A natural video object is encoded in one or more video object layers. Each layer enhances the temporal or spatial resolution of a video object. For single layer coding, only one video object layer exists.

Each video object layer contains a sequence of 2D representations at different time intervals referred to as video object planes (vops).[4] The traditional picture coding as in MPEG-2 can be thought of as coding a single rectangular object.

Video object planes are further divided into macroblocks of size 16×16. In contrast to MPEG-2, only the 4:2:0 chroma format is supported. Accordingly, each macroblock is encoded in 6 blocks. To obtain a macroblock structure from an arbitrary shaped vop the bounding box of the vop is calculated and extended to multiples of the macroblock size.

4.2.2 Compression

Again, compression efficiency is based on temporal and spatial redundancies. Hence, the concept of intra and inter coding is used in terms of I-,P- and B-vops. Consequently, the first vop is encoded in intra-frame coding mode (I-vop) and consecutive vops are encoded using inter-frame prediction (P- and B-vops).

In general, the coding techniques derived from MPEG-2 are applied to video object planes.

[4] According to MPEG-2, vops can be grouped in groups of video object planes

However, the motion compensation process and the DCT-based coding in MPEG-2 are block based. To cover arbitrarily shaped vops, special treatment of blocks on the border of a vop is necessary. Those blocks contain both pixels that belong to the video object and pixels outside the video object.

While blocks inside the vop are encoded using coding techniques described above, padding techniques, polygon matching motion estimation and shape adaptive DCT are applied to blocks on the vop border.

In addition to motion and texture, shape information is encoded. MPEG-4 visual adopted lossless and lossy techniques for shape encoding based on binary or gray scale shape information. While binary shape information provides only information which pixels belong to a video object, gray scale information defines transparency values (alpha values).

4.2.3 Scalability

Currently, MPEG-4 supports temporal and spatial scalability. In general two enhancement types can be discriminated: (1) The enhancement layer increases the resolution of a particular object or region of the base layer, (2) the enhancement layer increases the resolution of the entire base layer.

While in the case of temporal scalability both rectangular and arbitrarily shaped video object planes are supported, spatial scalability is presently restricted to rectangular vops. Therefore, MPEG-4 currently supports frame-based temporal, object-based temporal and frame-based spatial scalability.

4.2.4 Error resilience

Error resilience tools provided by MPEG-4 are divided into three major areas: resynchronization, error concealment and data recovery.

The first two techniques are already covered in the Section about error resilience in MPEG-2. However, MPEG-4 provides enhanced resynchronization techniques. The slice approach in MPEG-2 described above is based on the spatial dimension. After a group of macroblocks a resynchronization marker is inserted into the encoded data stream. However, due to the nature of the encoding process these markers are unevenly spaced throughout the data stream. Therefore, certain portions of the stream are prone to errors.

The periodic synchronization approach adopted by MPEG-4 solves this problem: The data stream is subdivided into video packets of a given length. The length of video packets is based on the number of bits contained in that packet not on the number of macroblocks. A resynchronization marker is inserted at the start of a new video packet.

After synchronization has been re-established, tools can be applied to recover data that in general would be lost. One particular recovery tool that relies on error resilient encoding of the data is reversible variable length codes (RVLC). In this approach, the variable length code words are designed such that they can be read both in the forward as well as the backward direction.

4.3 Generation of video object planes

As described in the previous Sections MPEG-4 visual provides coding techniques for encoding video object planes and content-based access to video objects in the scene. Therefore, image segmentation is crucial for exploiting the functionalities of MPEG-4. However, the segmentation process is not part of a the normative standard.

An image segmentation algorithm divides an image into regions according to a given criterion. Ideally, this partition corresponds to the objects in the video sequence. However, the perfect automatic segmentation approach is not available. There are, according to Marr [Mar82], mainly two reasons: (1) In most cases it is impossible to formulate the exact goals of segmentation and (2) semantic properties of the object under study need not result in particular visual distinctions in the image. Furthermore, digital images are corrupted during image acquisition and transmission.

Consequently, segmentation techniques restricted to certain scenarios and applications have to be developed. In Section 6 we describe a real-time segmentation algorithm that relies on the special properties of a teleteaching lecture.

5 Real time transport protocol

Applying MPEG compression techniques to an image sequence results in a bitstream containing the encoded video data. However, to transmit a bitstream of arbitrary length over datagram networks it has to be partitioned into data packets of appropriate size.

In the following we discuss a transport protocol for typical multimedia communication scenarios and applications. Such scenarios are for example audio and video conferencing sessions where several participants are connected via a network which provides unreliable multicast services. Each participant can send real time data and joins and leaves the session dynamically.

The Real Time Transport Protocol (RTP) is an application layer transport protocol which has been especially designed for transporting data streams with real time characteristics such as video and to "loosely" control sessions such as video conferences. RTP has been developed by the Audio-Video-Transport-Group (AVT), a special interest group of the Internet Engineering Task Force (IETF). Its development has been triggered by the joint interest of the group to provide an open interface for exchanging audio and video data over datagram networks such as the Internet. In order to send real time video over the Internet two services have to be provided:

(1) As mentioned above the stream has to be divided in small packets which fit in a datagram. This process is called framing [Cla90]. RTP provides a standardized packet format which is divided into a header part and a payload part. While the header part provides meta information such as timestamps, sequence numbers and data type identifiers the payload contains the essential data. RTP is open to transport any kind of media and therefore a payload format definition is necessary for each type of media. These payload format definitions are given in additional documents. Section 5.2 explains header and payload formats in more detail.

(2) RTP is typically run on top of unreliable protocols like UDP to make use of multicasting services. In order to monitor the quality of service of the underlying network and to give feedback about the participants of a (multicast) session RTP includes a control protocol called Real Time Control Protocol (RTCP). Consequently, a RTP session consists of two streams: The data stream and the control stream. In case that UDP is used as underlying transport protocol applications typically use even port numbers for the data stream and the next higher odd number for the control stream. Section 5.1 summarizes the services provided by RTCP.

RTP is an open protocol which can be used in many applications with different types of data, e.g. live Internet audio/video conferences or Internet TV. The core protocol is defined in Internet draft [Sch97] which revises RFC 1889[Sch96b][5]. This document describes protocol specifications which are common in all applications. Additional specifications for a particular application are given in separate documents, which define an application *profile* and one or

[5] Note that among other changes the draft specify protocol extensions for layered media streams.

several *payload format* specifications. The *profile* specifies extensions and modifications of RTP and defines payload type codes in order to identify the payload format. For example a RTP datagram with the payload type value 100 in the RTP header is mapped to MPEG-1/MPEG-2 streams. A profile for audio and video can be found in RFC 1890 [Sch96a]. The *payload format* specification defines how a particular payload (e.g. MPEG-1/MPEG-2) is to be carried in RTP. There already exist several Internet drafts which define payload format specifications for particular media streams. For example a payload format for MPEG-1/MPEG-2 can be found in [Hof97].

5.1 RTP control protocol

RTCP defines control packets which are periodically transmitted from each participant to the other participants of the session and performs two mayor tasks:

(1) It provides feedback on the quality of service of the underlying network. These informations can be used to allow flow and congestion control functions. E.g. a participant in a video conference can reduce his frame rate if the other participants report high packet loss rates.

(2) It allows the transmission of minimal session control information, e.g. the name and the email address of a participant.

5.2 RTP data transport

It is beyond the scope of this paper to discuss all profiles and payload formats in detail. Instead we first describe the RTP-header common to all payloads followed by an overview of the MPEG-1/MPEG-2 payload format as an example for other payload types.

The RTP datagram header contains information common to all payload formats. In Table 1 the format of such a RTP datagram header is described.

<table>
<tr><td colspan="10"></td><td>*1*</td><td colspan="9"></td><td>*2*</td><td colspan="9"></td><td>*3*</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">flags</td><td>M</td><td colspan="7">PT</td><td colspan="16">sequence number</td></tr>
<tr><td colspan="32">timestamp</td></tr>
<tr><td colspan="32">synchronization source (SSRC) identifier</td></tr>
<tr><td colspan="32">contribution source(CSRC) identifier
. . .</td></tr>
</table>

Table 1: Fixed RTP Header Fields

The first eight bits of the RTP header are used as **flags** and contain various informations like the version number and padding bits. The marker bit **M** is interpreted differently in different payload types and is followed by the payload type identifier **PT**. The **sequence number** is used to identify packet loss and to restore the original packet order. The **timestamp** reflects the sampling instant of the data transported within the RTP packet according to the Network Time Protocol. The next header field **SSRC** is intended to used as a unique identifier for a participant of a session which is chosen randomly by each participant. For the rare case that two participants choose the same SSRC the protocol describes algorithms to detect and handle such a collision. The contribution source identifiers **CSRC** are used in order to identify all contributors if data of several participants has been mixed together in the payload. For example in an audio conference one of the participants connects via a low bandwidth connection. In order to reduce the network load a gateway application can be used which “mixes” the data of several other participants in a single packet.

5.2.1 MPEG-1/ MPEG-2 payload format specification

Because often unreliable transport protocols are used packet losses may occur frequently. Furthermore participants may dynamically join and leave a session. Internet draft [Hof97] describes payload formats for MPEG-1 and MPEG-2 video streams which are defined with the intention to handle these situations gracefully. For example MPEG pictures can become quite large (in the case of I-frames) and a single picture is usually spread over several packets. Hence the payload defines fragmentation rules which guarantee that the MPEG stream is split at crucial points, e.g. at the beginning of a new picture. Furthermore the payload defines a header which contains important meta information about the stream, e.g. the frame number (within the current GOP) and several flags which are set if the packet contains the start of a new picture, a new slice or if MPEG parameters (e.g. frame size) are provided. That way new participants can easily detect packets in the stream which contain important meta information necessary for decoding the pictures by parsing the RTP header.

Table 2 summarizes the MPEG specific RTP header in the payload in order to provide a more practical sense for the abstract description in the previous paragraph.

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td><i>1</i></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td><i>2</i></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td><i>3</i></td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="5">MBZ</td><td>T</td><td colspan="10">TR</td><td>A
N</td><td>N</td><td>S</td><td>B</td><td>E</td><td colspan="3">P</td><td>F
B
V</td><td colspan="3">BFC</td><td>F
F
V</td><td colspan="3">FFC</td></tr>
</table>

Table 2: MPEG specific RTP Header Fields.

The first 4 bits **MBZ** are currently unused. They are reserved for future specifications. The **T** bit specifies the MPEG type. It is set if the RTP packet contains MPEG-2 data and erased if MPEG-1 is transmitted. The next ten bits define the temporal reference **TR** of the current picture relative to the current GOP, followed by several flags: The Active N flag **AN** is only valid for MPEG-2. Together with the new picture header flag **N** it signals changes of the MPEG-2 picture format. The **S** bit is set if the packet contains a new sequence header followed by the **B** bit and the **E** which signal the beginning or respectively the end of a slice. These bits are useful for the decoder if a packet loss occurred. In that case the decoder can easily skip packets until a necessary header is reached. The remaining bits signal information about the picture type and coding, e.g. if it is an I, B or P frame.

6 Applications

In this Section we summarize our integration of several techniques mentioned above into the well-known video conferencing tool vic [McC95]. In particular, we implemented the transmission of MPEG-2 compressed video [Sch98] and integrated a real-time segmentation algorithm for the generation of video objects in the context of teleteaching scenarios [Wid98].

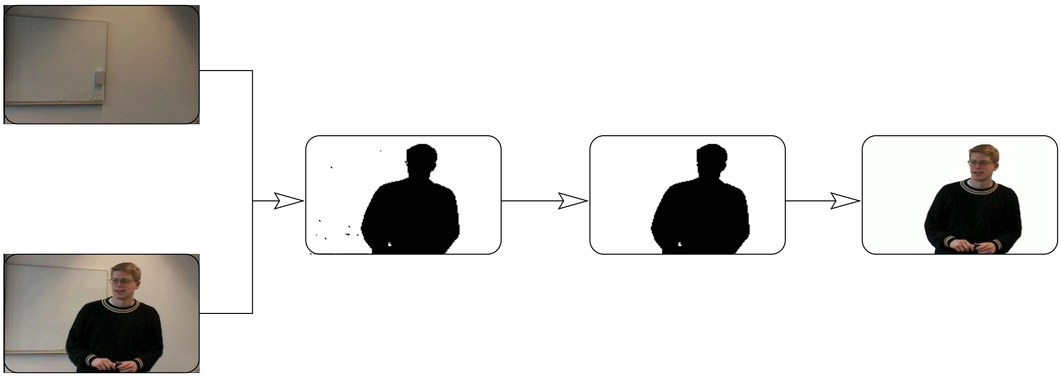

Figure 12: Real-Time Image Segmentation. In the first step an image is subtracted from the reference image. Then erroneously detected pixels are removed from the binary shape image. Finally, a video object plane is constructed by mapping the video frame to the binary shape.

6.1 Transmission of MPEG-2 encoded video

The implementation of the MPEG-2 transmission scheme relies on the principles already described in Section 5. The MPEG-2 stream is split up into packets according to the MPEG-2 RTP payload specification. The RTP packets are transmitted via UDP multicast.

A MPEG-2 decoder based on the reference decoder version 1.2 of the MPEG Software Simulation Group is integrated into the video conferencing tool vic. It assembles the RTP packets until a full picture is received and decodes the frame for displaying in the vic output window.

6.2 Generation of video objects

In a typical teleteaching session, three data streams are transmitted via multicast: (1) A video stream containing the speaker, (2) a corresponding audio stream and (3) educational material (slides etc.) presented using an electronic whiteboard [Eff97].

Normally, the speaker is the only video object of interest in the video sequence, while the background remains static and provides less information. Therefore, to reduce the size of the compressed video sequence it is useful to solely encode the speaker.

In the following we shortly describe a real-time segmentation approach for extracting foreground objects in front of a static background scene. The concept of this approach is closely related to the “blue screening” technology used in TV or in movies where e.g. an actor plays in front of a blue background. The blue background can be deleted and replaced by something else e.g. a steep abyss. Often this trick is used in news casts in order to place a picture or a sign in the background of the speaker.

Our approach follows the idea of blue screening in the following way: We first create a reference image of the background without any foreground objects. Afterwards each image of the video is “subtracted” from the reference image by comparing each pixel of the image with the related pixel of the reference image. If the difference between the two images is larger than a certain value *d* the pixel is considered to belong to a foreground object. In order to remove erroneously recognized pixels and small regions the image is filtered by an erosion and a dilation filter [Jäh97]. Figure 12 illustrates the steps of the process.

At the moment the transmission of the generated video object relies on the vic standard compression technique Intra-H.261 [McC95]. In the near future, we will integrate MPEG-4 support into the videoconferencing tool to exploit the functionality of the emerging standard.

7 Conclusions and Outlook

In this paper we described the MPEG compression standards for encoding natural video and appropriate transmissions schemes over the Internet. We discussed hierarchical encoding techniques that receives considerable attention due to the present heterogeneity of networks. Finally, we introduced prototype applications that integrate the basic concepts of object-based video transmission in an early stage.

Because of their capabilities to handle layered video streams MPEG-2 and especially MPEG-4 are promising frameworks for Internet-based multimedia applications. However, real-time encoding of MPEG compressed streams is computationally expensive and therefore currently demands specialized hardware. Furthermore, the error resilience techniques and transmission schemes have to prove their robustness in future applications.

We plan to extend the above mentioned applications with respect to the emerging MPEG-4 standard. Future work may comprise if MPEG-4 can fulfill the expectations placed in it.

References

[Ami96] Elan Amir, Steven McCanne, and Martin Vetterli. A Layered DCT Coder for Internet Video. In *IEEE International Conference on Image Processing ICIP '96, Lousanne Switzerland*, pages 13 – 16. IEEE, September 1996.

[Bur83] Peter Burt and Edward Adelson. The Laplacian Pyramid as A Compact Image Code. *IEEE Transactions on Communications*, 1983.

[Cla90] D. Clark and D. Tennenhouse. Architectural Considerations for a New Generation of Protocols. In *SIGCOMM Symposium on Communications Architectures and Protocols, Philadelphia*, pages 200 – 208. IEEE, September 1990.

[Dee89] S. Deering, C.Partridge, and D. Waitzmann. Distance Vector Multicast Routing Protocol. Internet Request For Comments, IETF, RFC-1075, August 1989.

[Dee91] S. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford Univerity, California, USA, 1991.

[Doe96] Peter K. Doenges, Tolga K. Capin, Fabio Lavagetto, Joern Ostermannand Igor S. Pandzic, and Eric D. Petajan. MPEG-4: Audio/Video & Synthetic Graphics/Audio for Mixed Media. *Image Communication. Special Issue on MPEG-4*, 1996.

[Eck97] Andreas Eckert, Werner Geyer, and Wolfgang Effelsberg. A Distance Learning System for Higher Education on Telecommunications and Multmedia - A Compund Organisational, pedagogical and Technical Approach. In *ED-MEDIA/ED-TELECOM'97, Calgary, Canada*. AACE Association for the Advancement of Computing in Education, 1997.

[Eff97] W. Effelsberg. Das Projekt TeleTeaching der Universitäten Mannheim und Heidelberg (in German). In teleteaching, distance education, mbone [Eck97].

[Gol93] E. Bruce Goldstein. *Sensation and Perception*. Wadsworth, Belmont, 1993.

[Hof97] Don Hoffmann, Gerard Fernando, Vivek Goyal, and M. Reha Civanlar. RTP Payload Format for MPEG1/MPEG2 Video. Internet draft, IETF, Audio/Video Transport Working Group, draft-ietf-avt-rtp-new-00, November 1997.

[ISO93] ISO/IEC 11172-2. Information technology – Generic coding of moving pictures and associated audio for digital storage media at up to 1,5 MBits/s – Part 2: Video, 1993.

[ISO95] ISO/IEC 13818-2. Information technology – Generic coding of moving pictures and associated audio – Part 2: Video, 1995.

[ISO97] ISO/IEC 14496-2. Information technology – Coding of audio-visual objects: Visual. Committee Draft, October 1997.

[ITU90] ITU. *Recommendation H.261 Video Codec for Audiovisual Services at p× 64kbit/s*. International Telecommunication Union ITU, 1990.

[Jäh97] Bernd Jähne. *Digitale Bildverarbeitung*. Springer Verlag, 4^{th} edition, 1997.

[Mar82] David Marr. *Vision*. Freeman, 1982.

[McC95] Steven McCanne and Van Jacobson. vic: A flexible Framework for Packet Video. In *MultiMedia '95 (San Francisco)*, New York, November 1995. ACM, ACM Press.

[McC96] Steven McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, Ca, USA, 1996.

[Mer97] Michael Merz, Konrad Froitzheim, Peter Schulthess, and Heiner Wolf. Iterative Transmission of Media Streams. In *Proceedings of the conference on Multimedia '97*, pages 283–290. ACM, 1997.

[Pen93] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[Poy96] Charles A. Poynton. *A Technical Introduction to Digital Video*. John Wiley & Sons, January 1996.

[Sch96a] Henning Schulzrinne. RTP Profile for Audio and Video Conferences with minimal control. Internet Request For Comments, IETF, RFC-1890, January 1996.

[Sch96b] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications. Internet Request For Comments, IETF, RFC-1889, January 1996.

[Sch97] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications. Internet draft, IETF, Audio/Video Transport Working Group, draft-ietf-avt-rtp-new-00, December, expirering date June 5^{th}, 1998 1997.

[Sch98] Armin Schieber. Implementation of the RTP payload for MPEG-2 video and integration into the videoconferencing tool vic (in german). Master's thesis, Lehrstuhl für Praktische Informatik IV, University of Mannheim, 1998.

[Tek95] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall Signal Processing Series, 1995.

[Wid98] Jörg Widmer. Development of image segmentation techniques for recognizing moving objects in front of static background and integration into the videoconferencing tool vic (in german). Studienarbeit. Lehrstuhl für Praktische Informatik IV, University of Mannheim, 1998.