

TFE department. Umeå University. SWEDEN

MPEG-4 style object-based codec with MatLab

Abstract

Video is nowadays much more than an instrument of leisure. It is used in fields like Medicine and Security in which it demands both quality and manageability. Compression plays here a key role: a good compression should supply easy storage and transmission without loss of information.

Many methods have been developed and some standards published to ensure the compatibility between systems. In the standard MPEG-4 an attempt is made to modify the traditional schema of an encoder by a content-based approach of compression. Multimedia applications would benefit from the handiness and the efficiency of compression would be definitely increased.

Several developers claim to use MPEG-4. Not all the profiles of MPEG-4 require object-based encoding, thus only a few of them, or maybe none, really perform it. Mixtures and combinations of different techniques partially based in the video content are used but the indications of the standard strictly speaking are not followed. In this project, an overview of video compression and its state of the art is first given, followed by an extended description of Object-based encoding steps. An implementation of an encoder with MatLab is then provided as a didactical approach to the problem. The application does not produce a MPEG-4 compliant bit stream and is then called Mpeg-4-style encoder instead of MPEG-4.

ABSTRACT	3
I. INTRODUCTION	7
II. OVERVIEW OF VIDEO COMPRESSION	9
II.1 TEMPORAL REDUNDANCY REDUCTION	11
II.2 SPATIAL REDUNDANCY REDUCTION	11
II.21 DCT	12
II.22 DWT	12
II.23 Quantisation	13
II.24 Reordering	13
II.3 ENTROPY ENCODING	14
II.31 Huffman coding	14
II.32 Arithmetic coding	14
III. MOTION ESTIMATION AND COMPENSATION	17
III.1 MOTION ESTIMATION	17
III.11 Cost function	17
III.12 Search method	18
III.13 Multipicture motion estimation	21
III.2 MOTION COMPENSATION	22
IV. DWT	25
IV.1 SUBBAND	25
IV.2 WAVELET TRANSFORM AND DISCRETE WAVELET TRANSFORM	25
IV.21 Definition	26
IV.22 Standard wavelets	26
IV.3 APPLICATIONS	28
V. MPEG STANDARDS	31
V.1 MPEG-1	31
Subsampling	31
Motion estimation and compensation	31
Spatial Transformation	32
Quantization	32
Table3. Quantization table for intra blocks	32
Entropy encoding	32
V.2 MPEG-2	32
Subsampling	33
Motion estimation and compensation	33
V.3 MPEG-4	33
Spatial Transformation	33
Entropy encoding	34
Levels and Profiles	34
VI. OBJECT BASED CODING	35
VI.1 VIDEO OBJECTS PLANES	35
VI.2 OBJECT-BASED VIDEO CODING PROCESS	36
VI.21 Shape Coding, alpha mask	36
VI.22 Foreground Coding	39
VI.23 Background coding	41
VI.3 SEGMENTATION	42
VII. MATLAB IMPLEMENTATION OF AN MPEG-4 STYLE CODEC	45
VII.1 ALGORITHM	45
VII.2 CODE	7
VII.3 RESULTS	8
VII.4 CONCLUSIONS	13
VIII. CONCLUSION	15
REFERENCES	17

I. Introduction

Video is nowadays used in several different applications, not only leisure. Video Conferences, Medical diagnostic, Security devices: daily tools more and more valuable but requiring an improvement in the means. Video compression is needed to facilitate both storage and transmission in real time.

Several compression procedures are developed and combined every day. To control the vertiginous growth of such an amount of techniques and ensure compatibility between them, standardization is essential. The Motion Picture Expert Group started seeking it out in the middle of the 1980s and is still actively working nowadays. After publishing MPEG-1 and MPEG-2 standards, the first version of MPEG-4 was brought out in 1998. An interesting innovation was the principle of content-based encoding allowing manipulation of the constituents for multimedia applications.

The Core and Main profiles of MPEG-4 Part2/Visual include tools to perform object-based encoding. The guidelines of this technique are extensively presented in the standard and a lot of research in this field has been done. However, the attempts to spread it have not been so successful yet as some points still need to be optimized.

Several developers claim to use MPEG-4 but only a few of them, or maybe none, really provide all the profiles. Mixtures and combinations of different techniques are blend and the real essence of the object-based encoding is blurred. In this project, after giving an overview of video compression and presenting the state of the art, an attempt to compile and recombine all the available information about the specifications of the standard's object-based profiles is done. An implementation of an encoder with MatLab is then provided as a didactical approach to the problem.

II. Overview of Video compression

A revolution has broken out in the media industry in the last decade. All techniques allowing improvements in the audiovisual field are welcomed and the research in audio and video material has nowadays taken a fundamental position in technology.

An essential area in which engineers are sparing no effort is Video Compression. Video Compression makes it possible to use, transmit, or manipulate videos easier and faster.

Many applications benefit from video compression. Compression is the process of converting data to a format that requires fewer bits. If there exists a loss of information along this process, the compression is lossy. Lossless compression is a manner in which no data is lost. The compression ratio obtained is usually not sufficient and lossy compression is required. In this report only lossy compression will be described.

The main goal in Video Compression is to minimize the weight of the files and maximize the quality of the reconstruction. The principle for achieving efficient compression is to eliminate unnecessary data. Two main features can be examined to deem the negligible information we can go without: the redundancy of data and the deficiencies of human visual system.

By redundancy of data, spatial, temporal and frequency point of view are meant. Indeed, analysing a video sequence reveals that a large amount of data is recognized to appear repeatedly. Within a single frame, large areas of pixels are homogeneous and present significant correlation. When analysing consecutive frames a big amount of redundant data between frames also exists. Hence a percentage of information can be discharged.

When elaborating the television systems, advantages have been taken of the deficiencies of the human visual system to simplify some of the elements: on the basis of these deficiencies, the number of frames per second or lines per frame has been adjusted, some colour corrections have been skipped, and some spectrum overlapping made possible. In Video Compression, once more, the eye and brain could be deceived and the irrelevant information will be eliminated.

Many compression techniques have been developed in the last decades. They can differ on the way the information is processed, the algorithms chosen or the coding methods used, but a general block-diagram (Figure 1) can be built to represent the philosophy of most of the video encoders.

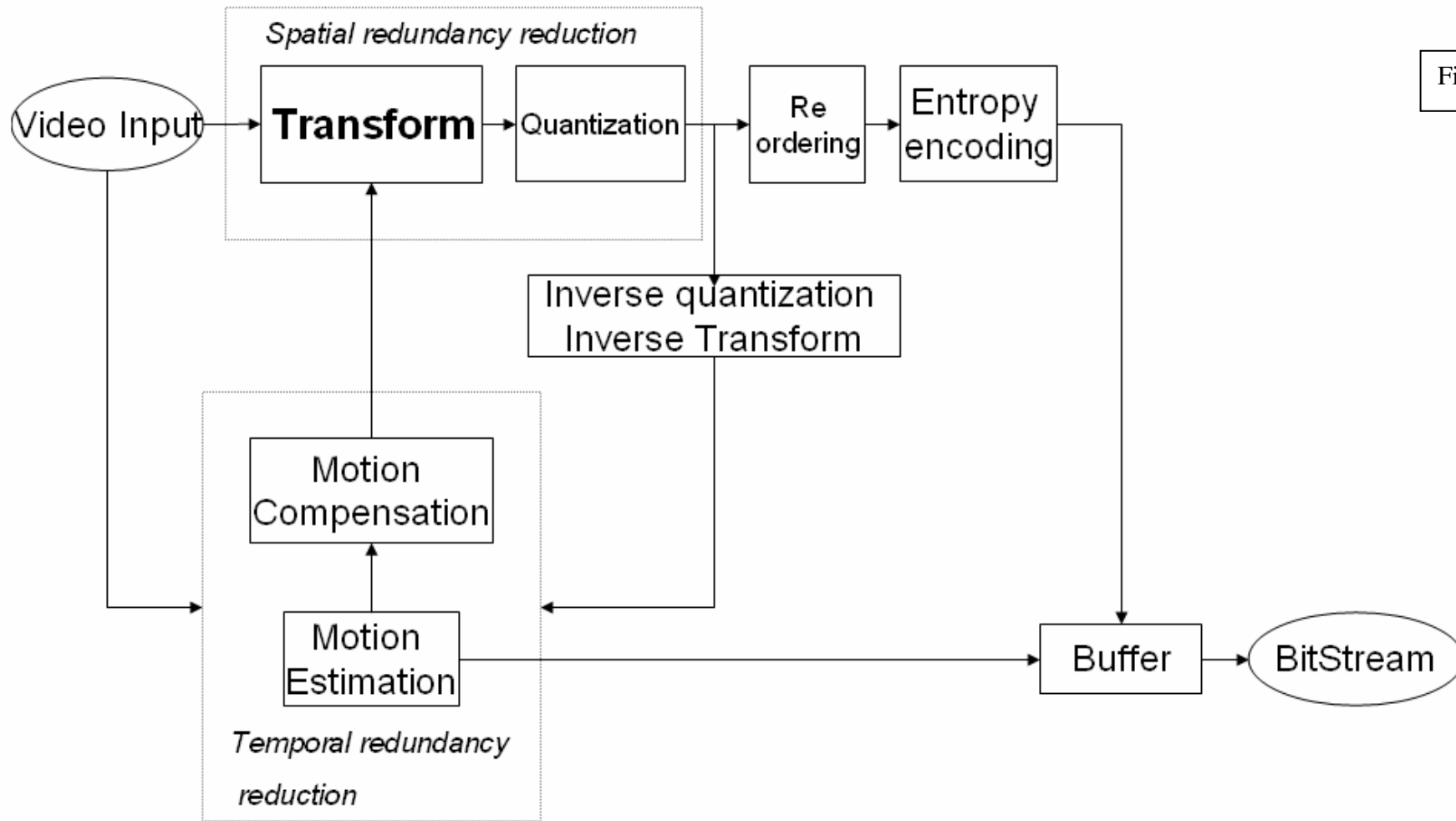


Figure 1.

Figure1. Block Diagram of a traditional encoder.

The main blocks are listed below:

- Temporal redundancy reduction
 - Motion estimation
 - Motion compensation

- Spatial Redundancy
 - Transform (DCT, DWT)
 - Quantisation
- Re-ordering and Entropy encoding

A fast overview of each block is now presented.

II.1 Temporal Redundancy Reduction

When analysing any series of consecutive frames from a video sequence, it is easy to recognize that, from one frame to another the content of the image has been modified in a really lightly way. The image information's difference from one frame to the other is normally reduced to some small changes due to motion or illumination alterations in the content.

This redundancy in the information will allow a reduction of the data. The concept of using a reference frame to code an upcoming one and thus minimize the amount of data is the key of the interframe predictive coding. The process can be described as a two steps process: first a prediction of a frame is made in base of a reference frame (in most of the cases a single temporally previous frame). Then, the difference between the actual current frame and its prediction is computed and coded in order to be transmitted. The parameters used in the prediction are also transmitted. The prediction step is associated to motion estimation. Motion compensation follows motion estimation and is the action of computing the residual. Both motion estimation and compensation methods are explained further in section III.

At the encoder a loop allows to recover each frame and reuse it as a reference for posterior frames to be encoded. It realises the inverse functions of the quantisation step and transformation step.

If the images are coded individually, time is not included in the compression process, and the coding is intra. For each frame, traditional static image compression methods can be used but the result is just a succession of images where no temporal redundancy is exploited. If the time factor is involved, a frame is coded based upon another one and the coding is named inter.

II.2 Spatial Redundancy Reduction

The residual frame obtained from the previous block will be processed to reduce spatial redundancy. This step involves a transform coding to convert the image into another domain. The transform should fulfil some criteria [23]

- the data in the transform domain should be decorrelated and compact
- the transform should be reversible
- the transform should be computationally tractable.

Two transforms are the most popular ones when speaking about video compression: a block based one -the Discrete Cosinus Transform- and an image-based one -the Discrete Wavelet Transform both of them will be introduced briefly here. A more extended description of DWT is found in IV.

II.21 DCT

The Discrete Cosinus Transform is the most common transform used in video compression. It is an orthogonal transform not too computational demanding (fast algorithms are available) and which inverse can be easily calculated. For highly correlated image data, the DCT provides an efficient compaction and has the property of separability. The two dimensional processes consist in transforming an original N by N block of pixels from the spatial domain to the DCT domain. If Y is the forward DC Transformed of an N by N block X, it fulfils:

$$Y_{xy} = \frac{2}{N} C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(2i+1)x\pi}{2N} \cos \frac{(2j+1)y\pi}{2N}$$

$$\text{where } C_a = \begin{cases} \sqrt{\frac{1}{2}} & \text{if } a=0 \\ 1 & \text{Otherwise} \end{cases}$$

Y is a set of N by N coefficients representing the data in the transformed domain. A set of waveforms is defined for each possible value of N (usually N=8 thus there exists 64 waveforms). Each coefficient can be seen as the weight of each of these basis patterns or waveforms. By summing all the waveforms scaled by the corresponding weight the original data can be recovered.

II.22 DWT

Even if DCT presents several advantages and is the main transform used in image and video compression, there exists a clear drawback: because DCT is block based, blocking artefacts will appear. The objective in subband coding and wavelet transform is to transform the image in a different way so that the result will be block artefact free.

The basic operation in wavelet transform is to filter an image with a low pass filter and a high pass filter and down-sample the output by a factor of 2. Both operations on the x direction. Two new images are obtained L and H. They are filtered and down-sampled again but this time in the y direction. Four subbands images are obtained which can be combined to recover the original one. The same amount of information is present, but this new configuration is more suitable for efficient coding.

II.23 Quantisation

After transformation the whole information about the image and the whole amount of data is still hold, no compression has been performed yet. However, the transformation makes the energy to be distributed in a more easily reducible way. For DCT coefficients, the maximum of energy is concentrated at the lowest frequencies components and thus the majority of coefficients have little energy. By applying quantisation, insignificant values will be removed. The principle of quantization is to divide the values by a nonzero positive integer (quantization value) and round the quotient to the nearest integer. Uniform quantizers are usually defined by two parameters: the step 'q' and the factor 'S'.

Human vision deficiencies can be used to make the quantisation even more effective. For high frequencies the human eye is less sensitive to distortions and then coarser quantisation (larger step size) can be applied. [23]

Variable Uniform Quatization.

Different variations of the uniform quantiser are used in video compression. The most popular one is the Variable Uniform Quantizer (VUQ). It is applied by means of a quantizing lookup table that provides quantization step sizes and multipliers that scale the coefficients to smaller values. [5]

For a matrix N by N of coefficient to be quantized, the values for the step size 'q' are determined from NxN matrices such that each coefficient position can have a different value of q.

Prior to the step of encoding those quantised coefficients, a reordering should be performed to ensure an effective entropy encoding.

II.24 Reordering

DCT coefficients should be reordered prior to Variable Length encoding. A typical way of reordering the coefficients of a block is zigzag scanning. It allows obtaining an array of data with the non-zero coefficients regrouped at the beginning and a sequence of zeros at the end.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Table 1. Order in which the coefficients are scanned for reordering in the zig-zag scanning.

For DWT coefficients a zero-tree encoding algorithm is used instead. Further explanation can be found in section IV.3

II.3 Entropy encoding

To reduce the bit rate, the quantised transformed coefficients are coded. Entropy encoding encodes the data according to the information content. Frequently occurring messages, carrying more information, are coded with longer words, while less common messages, carrying less information, are coded with shorter words.

Any Variable Length Coding (VLC) algorithm can be used. The two most popular VLC algorithms are Huffman coding and Arithmetic encoding.

II.31 Huffman coding

Huffman code achieves the shortest average possible code word length. Each symbol is assigned a VLC in base of the probability of occurrence of different symbols. For each symbol the probability needs to be calculated. A Huffman code tree is then generated by the following steps:

- order the data in increasing order of probability
- combine the two lowest probable data and assign to this new association the sum of their probabilities.
- restart the process including in the list of data this last association and its new probability instead of the two data with lowest probability.

The process is reiterated until only one single node remains. Then, the tree is traversed through its branches starting from the last node, and ending on each of the data. The branches on the path are indexed '0' or '1'. A code can thus be generated for each data by concatenating in a sequence the '0's and '1's came across.

Huffman coding provides an accurate approximation of the data but one disadvantage is that the decoder will need to get the information of the Huffman tree and this implies extra data to be transmitted. Furthermore the computation of the probabilities for large video sequence introduces a delay. Pre-calculated VLC tables can be used instead. In section IV.3 this philosophy, which is used in MPEG-4, is described.

II.32 Arithmetic coding

The principle is to represent the data by a single number, included in the range [0 1] instead of using codewords thus the coding is more efficient.

The idea behind arithmetic coding is to have a probability line, 0-1, and assign to every symbol a subinterval in this line. The size of each subinterval is proportional to the frequency at which it appears in the message. , the higher the probability, the larger the range assigned. When encoding a symbol, the current interval is divided into subintervals like in the previous step. This new interval is the basis of the next symbol encoding step.

Binary arithmetic coding resolves the problem that can arise when the final range becomes smaller than the precision of a computer.

The disadvantages are that the whole codeword must be received to start decoding and that if there is a corrupt bit the entire message can be corrupt.

Further details about Context based Arithmetic encoding are given in section VI.21.

With entropy encoding, the process of compression is almost done. The data to be sent -motion vectors, VLC quantized coded coefficients and additional information like parameters or algorithms used- will then be multiplexed so that a bitstream is formed.

Several compression methods have been developed, combining in different ways the blocks described here, adding new features, changing parameters or innovating with new tools. Each application has its own needs and by adapting the algorithm to each particular application the compatibility between systems can be endangered. Hence the decision to define some standards.

Before introducing those standards, an overview of Motion estimation and compensation and Wavelet transform will be presented.

III. Motion estimation and compensation.

A rudimentary video coding scheme includes a process to eliminate temporal redundancy.

Indeed, when speaking about most common video sequences, there exists a high level of redundancy between consecutive frames. Unless a shot is produced and a new and totally different scene is shown, the changes from one frame to the other are minimal. The idea of temporal redundancy reduction is to encode first a reference frame and for the consecutive frames encode only the difference between the reference frame and the current frame. This difference is seen as the error or residual. For static areas, this error is 0 and do not need further coding. The difference will be significant for areas with moving objects but also for frames with changes in illumination or camera effects (panning, zoom...). Instead of coding directly the difference between two frames, to reduce even more the amount of data to be coded, an estimation of the motion can be carried out. The image is then motion compensated and the error corresponds to the difference with this compensated image instead of the original not compensated image.

To obtain an accurate motion compensated image and thus a minimal residual, first the motion estimation has to be as good as possible.

III.1 Motion estimation

The problem of motion estimation described in the video coding standards is exposed by Bhaskaran [3]: given a reference picture and a M by N macroblock (MB) in a current picture, the objective of motion estimation is to determine the M by N block in the reference picture that better matches the characteristics of the macroblock in the current picture.

Several methods of motion estimation have been developed. They can be classified according to different criteria.

III.11 Cost function

The cost function describes the degree of matching between two macroblocks and depending on the characteristic chosen to compare the blocks. This characteristic is one of the criteria to group estimation methods. [3]

In the following equations, The pixels of the current MB are denoted $C(x+k, y+l)$ and the pixels in the reference picture are $R(x+i+k, y+j+l)$.

- MAE. It consists in minimizing the Mean Absolute Error $MAE(i, j)$ where

$$MAE(i, j) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

The coordinates (i, j) for which MAE is minimized define the motion vector.

- MSE. The Mean Squared Error (MSE) can also be chosen as a cost function.

$$MSE(i, j) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|^2$$

- The correlation between blocks is another way of comparing matching ratio.

- PDC. It is the Pixel Difference Classification. The principle is to count the number of matching pixels between two blocks. The goal is to maximize PDC.

$$PDC(i, j) = \sum_k \sum_l T_{i,j}(k, l)$$

where given a threshold t :

$$T_{i,j}(k, l) = 1 \quad \text{if } |C(x+k, y+l) - R(x+i+k, y+j+l)| \leq t \\ = 0 \quad \text{otherwise}$$

- BPDC and BPROP. The Binary Pixel Difference Classification (BPDC) is the binary representation of the PDC. In this index, all the matching pixels have same weight. If more weight is assigned to the most significant bits, the binary level matching criterion is called BPROP and the objective will be to minimize its value.

- DPC. The Difference Pixel Count Criterion is another variation of the BPDC in which instead of the matching pixels, the quantized matching pixels are evaluated.

- BPM. The Bit-Plane Matching Criterion. The pixels of both current and reference frame are assigned only one bit values. The MAE criterion is then applied in a simplified form. The idea under this method is to detect edges in order to describe the motion estimation.

A good advise given by Bhaskaran [3] is to combine different criteria: a first low-complexity matching criterion can be used to find a set of candidate motion vectors and a more accurate criterion can be use afterwards to refine the search.

Any of those matching criteria can be used with any search method. Search method is another way of classifying motion estimations.

III.12 Search method.

Motion estimation is one of the most computational demanding processes in compression. Several algorithms for matching blocks have been developed. We will see later, when speaking about compensation, that not only block based strategies are used when dealing with motion.

In his project, Barjatya [3] describes and evaluates the fundamental Block Matching Algorithms for Motion Estimation. Here an overview is given.

Ideally, the search of the best matching MB should be performed on the whole picture. But this is not possible in a practical point of view. The search is then restricted to a $[-p, p]$ search region around the original location of the MB in the current picture. 'p' is the search parameter.

Figure 2.

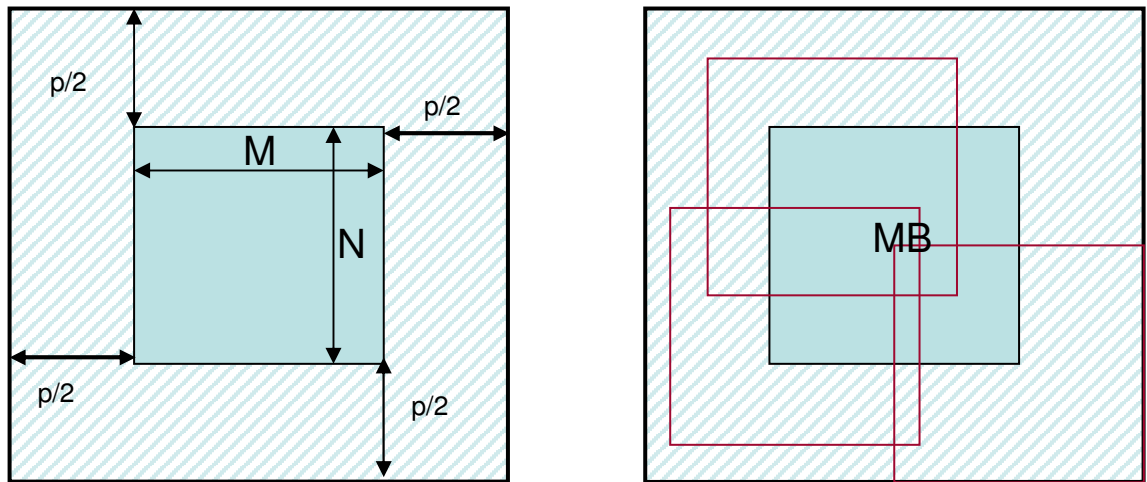


Figure 2. Search Region (left) and some MBs to analyse the matching with the reference MB. (right)

Exhaustive search (ES) or Full search

The cost function (MAE or other) is computed at each location in the search space. This algorithm guarantees finding the best matching, but it is the most computational expensive one. To reduce this complexity, the number of search location or the number of operations should be decreased (fast algorithms). This second solution implies not computing the cost function for all pixels in the MB.

Two-Dimensional Logarithmic Search (TDLS) and Three Step Search (TSS).

In Two-Dimensional Logarithmic search [3] the search rectangle is first divided into two areas: an internal and an external rectangle. Then, instead of computing the cost function for the whole rectangle $[-p/2 p/2]$, it is only computed at the location (0,0) and at eight locations around this one (major points of the analysed rectangle, distance between points: S). The best match location is then taken as the starting point for next search that will be performed in the same way but using a rectangle of half the size of the previous one. The process continues until the eight locations are spaced by one point. Then the best result in the cost function reveals the best match. According to Bhaskaran [3], the complexity of this algorithm is 3.3% the complexity of the extended search one.

Three steps search is a particular case of Two-Dimensional Logarithmic search in which 'p' is set to 7 (and so the process will have 3 steps before reaching the one-point distance locations) that is estimated sufficient for videoconferencing applications. Small motions are missed.

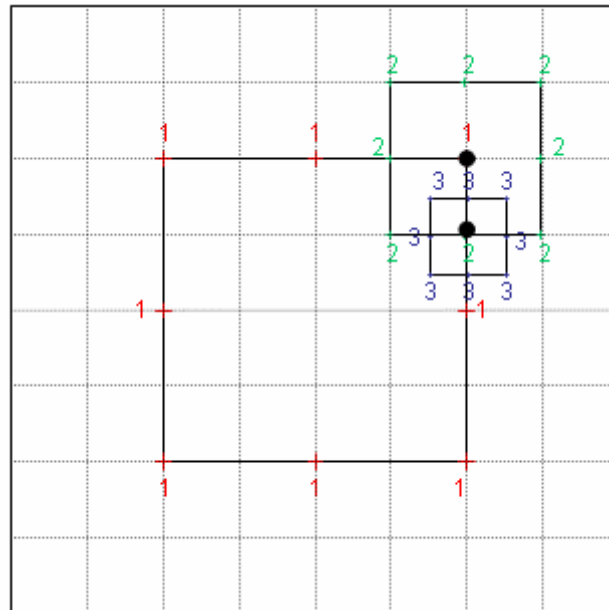


Figure 3.

Figure 3. TSS algorithm. First the positions labelled 1 are treated. the best match (here top right corner) is the centre for the new analysis of the positions labelled 2. The same process for number 3.

Parallel Hierarchical One-Dimensional Search (PHODS)

This algorithm is similar to TDLS, but with a particularity: the search is done independently along the two dimensions. The advantage is that the cost function computations are parallelizable that the flow of data is regular. [3]

New Three Step Search (NTSS)

This algorithm is similar to TSS but with the advantage that the process of searching the best origin can be stopped earlier. In addition to computing the cost function at the same 8 points as TSS ($S=8$, major points of the currently analysed rectangle), this computing is also carried out for 8 points 1-point distant ($S=1$) from the origin. In case the best match is at the origin, then the search can be stopped. Otherwise, if one of the locations one-point distant provides the lowest cost, the origin is moved to this point and check for weights adjacent to it. As the cost function has already been computed at some of the locations around this new origin, the number of operations will be reduced and the best match will be found quickly. If the best match is in one of the 8 points further, the TSS explained earlier is used but in this case, instead of decreasing the computational expenses, the NTSS will have supposed more operations.

Simple and Efficient Search (SES)

SES works on the assumption that there cannot be two minimums in opposite directions. The principle is similar to TSS but, in each step we will have two new phases. The search area is divided into four quadrants and in each one the same operation is performed: 3 points are selected (one in the origin and 2 at a distance S) and depending on their distributions, some few additional points are chosen in the second phase. As in TSS we look for the best match, move the origin and repeat the whole process until the one-point distance locations are evaluated. This process saves a lot in a

computational aspect, but the results in terms of best matching are not totally satisfactory.

Four Step Search (FSS)

The principle is close to the one for TSS again but no matter the value of p , s is always set to 2. The cost function is computed for 9 locations of a 5 by 5 window. As for NTSS if the centre is the best match, the algorithm stops, otherwise the centre is moved and the same process is repeated. In the last step, the window has a size of 3 by 3. The worst situation will be lightly better than NTSS.

Diamond Search. (DS)

The steps are exactly the same as in FSS but the pattern is not a square but a diamond instead. Two types of step can then be defined. This method is really accurate and useful to find global minimum.

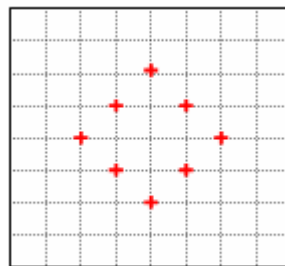


Figure 4.

Figure 4: Pattern for Diamond search.

Adaptive Rood Pattern Search (ARPS)

The principle is based on the fact that the direction of the general motion is usually coherent in the whole frame and so if the motion for MBs around the current analysed MB has already been computed, it can be used to calculate the current motion. By combining a prediction of the vector and a Diamond search, this algorithm allows saving a lot of computational time.

Some others algorithms are developed (i.e. Cross Diamond Search and its variations). The principle is almost always based on the same ideas, but some improvements are made. One of the key points to reduce computational time and expenses is to allow stopping the algorithm at any moment without having to perform useless operations. Hierarchical motion estimation is also a good strategy to base the new algorithms on. It consists in combining different search algorithms in a data adaptive way.

III.13 Multipicture motion estimation.

In the previous considerations, only one reference MB in a previous image is used to proceed to motion estimation. However, a better approximation of the global motion can be obtained by analysing several frames, more time-distant. This idea is

exploited in bidirectional motion estimation where both a forward and a backward estimation are computed.

Whatever matching criterion or the search algorithm, the parameters of the motion model are described and coded in the bit-stream. We should be aware not to make this part of the data become too large.

III.2 Motion Compensation

The current image has to be motion compensated to minimize the energy of the residual. This can be done in different ways, and here some of them are presented.

Global Motion Compensation (GMC)

Mainly camera motions are reflected by GMC, but moving objects are not so well modelled. The main advantage is that there are no blocking artefacts introduced.

Block-based Motion compensation (BMC)

Block based Motion Compensation is probably one of the most common way to achieve compensation. The idea is to compensate for movements of rectangular blocks. A search algorithm is used to identify the best matching block in a reference image and this area will be the predictor for the current block. The size of the block is often 16 by 16, corresponding to a macroblock, but sometimes 8 by 8 or 4 by 4 blocks are used. A variation of the BMC is the Variable block-size motion compensation (VBSMC) in which the size of the blocks used is dynamically adapted to the needs and the data

BMC is popular because it presents several advantages: it models precisely most of the usual motions found in video sequences and fits well with rectangular frames. Furthermore, it is not too computational expensive. However, for real objects, because their edges are not so neat, BMC is not the best solution, but the main disadvantage of BMC is that it introduces blocking artefacts in the borders.

Overlapped block motion compensation (OBMC)

OBMC tries to settle the problem of blocking artefacts by using overlapped blocks. each pixel belongs to 4 blocks, and so 4 estimation of it will be made. A pondered mean is then computed to obtain the final estimation free of blocking artefacts.

Sub-pixel motion compensation

It consists on using a prediction from interpolated sample positions in the reference image basing this idea on the fact that the best estimation can not be found by using integer pixels grids but by fractional pixel accuracy. [23] Interpolated pixels are computed between the real pixels of the image and the estimation is carried out upon those new pixels. The compensation will be improved but the complexity is high, and the main drawback is that the the information of the motion vectors to be transmitted will be heavy. A particular case of SPMC is Quarter Pel Motion Compensation.

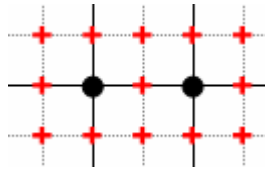


Figure 5.

Figure 5. Half-pel method. The crosses are the sub-pixel and the dots the pixels.

Region based motion compensation

The idea here is to perform the compensation of arbitrary shaped regions. This implies processing first the image to obtain the objects by segmentation and encoding the shape information.

IV. DWT

As explained earlier, Discrete Wavelet Transform coding has the advantage compared to DCT that it is blocking artefact free.

Following the structure used by Ghanbari[9] to explain Wavelet transform, we will first explain subbands to come to the description of Discrete Wavelet Transform.

IV.1 Subband

Subband coding is at the origin of Wavelet transform. The idea is to divide the spectrum of a signal into frequency bands and, after coding them, transmit each band separately. This principle suits perfectly image coding as images usually have most of their energy concentrated in the lower frequencies and as the deficiencies of the human vision prevent the spectator from differentiating distortions in both high and low frequencies.

When the image has been filtered by band pass filters, each resulting subband has a much narrower bandwidth and, according to Nyquist, can be subsampled. To obtain a perfect reconstruction afterwards, the filter should be ideal and should not present any overlapping. This is not practical and the gradual transition of the filters introduces aliasing in the process. To eliminate the aliasing component that is introduced, at the decoder after up-sampling and filtering, the bands are summed again. The key to avoid aliasing is to design the filters in such a way that the aliasing from one band will be cancelled with the one of another band. A perfect reconstruction is then expected.

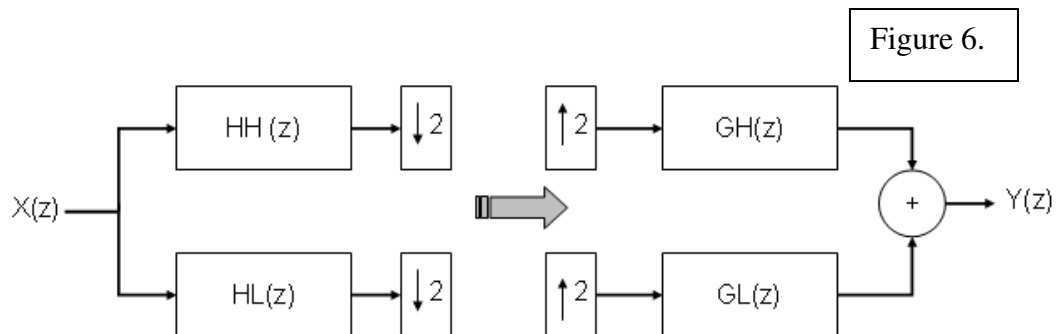


Figure 6. Subband block diagram. Analysis and Synthesis.
 $GL(z)=HH(-z)$ and $GH(z)=HL(-z)$ to have perfect reconstruction but it involves delay.

IV.2 Wavelet transform and Discrete Wavelet Transform

Subband coding is based on frequency analysis. Wavelet transform instead is based on the approximation theory. This theory was already used in Fourier expansion coming from the idea that a signal can be expressed as the sum of a series of sines and cosines. However in Fourier analysis only frequency resolution and no time resolution is considered. With Wavelets, both time and frequency variation will be recreated. [19]

IV.21 Definition

The definition of the wavelet transform is:

$$X_w(a,b) = \int_{-\infty}^{\infty} x(t)\Psi_{a,b}(t)dt$$

where a mother wavelet ψ is dilated a certain scale a and translated a certain time b to form the basis function Ψ defined by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right)$$

The wavelet transform of a one-dimensional function is two-dimensional; the wavelet transform of a two-dimensional function is four-dimensional.

For images, the discrete wavelet transform is used basing the approach on the fact that any square integrable function $x(t)$ can be represented as a linear combination of basis functions (Ghanbari [9]):

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \alpha_{m,n} \psi_{m,n}(t)$$

where $\alpha_{m,n}$ are the wavelet transform coefficients:

$$\alpha_{m,n} = \int_{-\infty}^{\infty} x(t)\Psi_{m,n}(t)dt$$

In other words, the wavelet transform will be a set of filters with coefficients equivalent to discrete wavelet functions. The concept of higher order systems is used to describe a more complex version of the one in figure 6 . Instead of one level decomposition, several stages can be repeated.

IV.22 Standard wavelets.

Wavelets in general are functions that can be used to efficiently represent other functions. When describing a wavelet two functions are given: scaling function ϕ (if orthogonal) and the wavelet function ψ . Usually a number next to the wavelet number represents the number of vanishing moments. Two important properties of wavelets are the admissibility and the regularity conditions. Some other properties which can be required when designing wavelets are orthogonality, compact support, rational coefficients, symmetry and smoothness. [6] [13]

Some typical wavelets and their properties will be listed here.

Haar wavelet

The haar wavelet is one of the most common used wavelets. It resembles a step function and is defined by:

$$\begin{aligned} \Psi(t) &= 1 && \text{if } t \in [0, 0.5[\\ \Psi(t) &= -1 && \text{if } t \in [0.5, 1[\\ \Psi(t) &= 0 && \text{otherwise} \\ \\ \varphi(t) &= 1 && \text{if } t \in [0, 1] \\ \varphi(t) &= 0 && \text{otherwise} \end{aligned}$$

The haar wavelet has all the following properties: orthogonality, compact support, symmetric scaling function, anti-symmetric wavelet function, and one single minimum.

It is the only one which is compactly supported, orthogonal and has symmetry.

Daubechies wavelets

The haar wavelet is included in a big family of wavelets: the Daubechies wavelets (dB). These wavelets have no explicit expression. They are orthogonal and have compact support. The properties of symmetry or the number of vanishing moments depend on the order N of the dB.

Shannon wavelet

It is constructed from the Shannon multiresolution approximation and fulfils:

$$\psi(t) = \frac{\sin 2\pi\left(t - \frac{1}{2}\right)}{2\pi\left(t - \frac{1}{2}\right)} - \frac{\sin \pi\left(t - \frac{1}{2}\right)}{\pi\left(t - \frac{1}{2}\right)}$$

Its properties: orthogonal symmetric scaling and symmetric wavelet function, infinite number of vanishing moments, infinite support and slowly decaying non-causal IIR filters.

Symlets wavelets

By applying some modifications to Daubechies wavelets, some more symmetry can be obtained. This results on the Symlets wavelets which have the following properties: orthogonality, compact support, nearly linear phase for φ .

Another variation of Daubechies wavelets is Coiflet wavelets. The symmetry is even more increased.

Cohen-Daubechies-Feauveau

The Cohen-Daubechies-Feauveau (CDF) wavelets are biorthogonal wavelets for which a number of moments are made zero.

Mexican hat function

The definition is:

$$\psi(t) = \left(\frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} \right) (1 - x^2) e^{-\frac{x^2}{2}}$$

It satisfies the admissibility condition and has two vanishing moments.

Several other wavelets have been developed, such as Meyer wavelet, Battle Lemarie wavelet, Morlet wavelet. More information can be easily found. [13][19]

IV.3 Applications

Wavelets can be used in several different applications for signal processing, detection of frequencies and discontinuities, noise reduction (i.e. digital audio denoising). [25] [19] [27] [29] [30]

When dealing with images, they can be used again for noise reduction, but also for object recognition, (Wavelet-based Semiautomatic Segmentation, i.e. edge detection), Motion compensation, contrast enhancement, texture analysis, and of course Compression.

Compression with wavelets.

The DWT can replace the DCT in the transform step of compression described in figure1. For image compression Daubechies wavelets or biorthogonal CDF wavelets are the most commonly used ones

The steps of quantization and VLC suffer some changes. If not too many levels of decomposition are used, there will exist residual correlations between pixels that can be reduced by DPCM coding. The higher order wavelet coefficients are quantised by successive approximation and afterwards exploitation of similarities of the bands. They will be then Embedded Zero Tree Wavelet (EZW) encoded.

The Embedded Zero Tree Wavelet (EZW) algorithm is described here briefly.[9]

- Compute mean of image
- An R stage wavelet is applied to the resulting zero-mean image
- An initial yardstick will have a length l of half the maximum absolute value of the wavelet coefficients.

- Three lists are generated: one determines the coordinates of the coefficient in the order they will be scanned, and two other empty lists (subordinate and temporary lists).

- The wavelet coefficients are scanned and assigned '0' if they are smaller than the current yardstick length, ' $\pm \frac{3l}{2}$ ' otherwise.

- Following to the order described in the dominant list, the reconstructed coefficients are scanned again, generating a string of symbols: the sign of a coefficient

is appended to the string, and its coordinates to the subordinate list. If a coefficient is zero, its coordinates are appended to the temporary list. More details are given in Ghanbari [9].

- The yardstick length is now set to $l/2$.

- The coefficients which previously have not been reconstructed as zero are scanned again according to their order in the subordinate list, and adding to them either $+l/2$ or $-l/2$ and the corresponding sign is appended. The subordinate list is reordered so that the coefficients whose reconstructed values have higher magnitudes come first. The '+' and '-' symbols of this pass are encoded with an arithmetic coder.

- The temporary list is emptied after replacing the dominant list.

The whole process can be repeated several times until the size of the bitstream is the desired one.

A header will include information necessary to the decoder: number of stages in the wavelet transform, image dimensions, initial value of the yardstick l , image mean...

A variation of EZW is the Set Partitioning In Hierarchical Trees (SPIHT). In SPIHT, the main difference is the way the subsets of the wavelet coefficients are partitioned. [9]

V. MPEG standards

The Moving Picture Experts Group (MPEG), created in 1990, is part of the International organization for standardization (ISO) and the International Electrotechnical Commission (IEC). The group appeared from the worry and need to store moving images and to establish some standards for this.

The first official researches resulted in the ISO/IEC 11172 norm or MPEG-1. The idea was to allow storage of video with the corresponding audio in CD-Rom supports with a transmission rate around 1.5Mb/s. The main improvement in compression efficiency compared to JPEG format was to take the advantage of temporal redundancy. MPEG-1 was just a first attempt to establish a standard, and the low quality pushes the group to create a more advanced standard, indexed as ISO/IEC 13818 or MPEG-2. MPEG-4 is the following standard described by the group as the intermediate level MPEG-3 was incorporated into the scope of MPEG2.

An MPEG standard describes the structure or syntax of a bitstream, the semantics of this syntax, and the basic process of compression, but it allows total freedom on the implementation method as long as the result fits the established syntax and so ensures the compatibility between products and systems. Reference softwares are available. To test the compatibility of an encoder, the bitstream obtained can be tried to be decoded with the reference software. However, this condition is compulsory but not sufficient.

Different set of characteristics can be found for each standard application, due to the fact that different combinations of tools and objects can be encountered. An object is a video element that is coded using one or more tools. It is unusual that a codec requires the implementation of all the tools available in a standard; levels and profiles are then defined. A profile defines a subset of coding tools and level defines constraints on the parameter of the bitstream. [23] [1][2][5][7][9][10] [11][16]

V.1 MPEG-1

MPEG-1 does not work with frames but with pictures as it does not recognize interlaced video. MPEG-1 algorithm follows the typical structure of any coder described in section I. Some particularities can be pointed.

Subsampling

Human eyes are more sensitive to luminance than chrominance. To reduce the amount of information to be encoded MPEG-1 subsamples both chrominance channels, the result is a 4:2:0 format.

Motion estimation and compensation.

MPEG-1 uses motion estimation and compensation. Three types of images are defined: I, P and B. I stands for Intra and designs pictures that are compressed using only information from the picture itself without any reference to other pictures. Predicted pictures (P) refers to pictures coded using motion estimation computed from past pictures (I or P). Finally B pictures involve a Bidirectional estimation using previous and future pictures. A group of pictures (GOP) is a set of pictures that has to contain at least one I-picture. Generally the first and last pictures are of type I.

Spatial Transformation.

Block based (8 by 8) DCT is applied.

Quantization

Two reference matrices are used for quantization: one for inter blocks and one for intra blocks.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Table2. Quantization table for inter blocks.

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

Table3. Quantization table for intra blocks.

Entropy encoding.

The smallest coefficient of the DCT is coded using a differential prediction method. The rest of coefficients are zigzag scanned and coded with Huffman as Variable Length Code.

The compression ratio obtained with this standard is over 100:1, but the quality is really low and in order to improve it, more research was made.

V.2 MPEG-2

In 1991 the group of experts started working on a new standard which would be more efficient and especially would provide a better quality than MPEG-1, with the additional requirement to be compatible with this one. In 1994 the MPEG-2 standard was published, with a bitrate increased to a range going from 4 to 9 Mbit/s, and interlaced video supported. A toolkit-like approach of the standard was adopted that will

allow defining profiles and levels depending on the application requirements. Seven profiles were defined: Simple, Main, 4:2:2, SNR, Spatial, High and Multiview.

MPEG-2 is really similar to MPEG-1 in most of the blocks: it is based on DCT coding, block based motion compensated with P and B frames and Huffman coded. The main difference and enhancement is the possibility to process interlaced video. A couple of new other features that make MPEG-2 unique are presented below.

Subsampling

In addition to the 4:2:0 format used in MPEG-1, MPEG-2 supports 4:2:2 and 4:4:4 subsampling formats.

Motion estimation and compensation.

The block matching for motion estimation is carried out by Sequential search, Logarithmic search or Hierarchical search algorithms. All motion vectors are specified at half-pixel resolution

With MPEG-2, higher resolutions, bitrates, and frame rates are expected.

Nowadays, MPEG-2 is the standard format used for satellite TV, digital cable TV, DVD movies, and HDTV. MPEG-2 is also commonly used to distribute video on the internet.

V.3 MPEG-4

The effort in MPEG-4 development was focused in obtaining flexibility, scalability, efficiency and robustness. MPEG-4 was published in 1998 with the aim of fulfilling the needs of multimedia applications. [8] [14][18] [21][22][23]

Two particularities of MPEG-4 are the transform and the VLC blocks.

Spatial Transformation.

An attempt to use DWT instead of DCT is made. Wavelet filters recommended are Daubechies (9,3) (Table 4).

<i>n</i>	lowpass	highpass
0	0.99436891104360	0.70710678118655
±1	0.41984465132952	-0.35355339059327
±2	-0.17677669529665	
±3	-0.06629126073624	
±4	0.03314563036812	

Table 4. Coefficient of analysis filters for Daubechies (9,3).

Entropy encoding.

As seen earlier in I.31 Huffman coding is not so efficient for a practical purpose with large video sequences. Huffman based coding is used instead in MPEG-4. It defines sets of codewords based on probability distributions of generic video. [23] Two examples of Huffman based coding are Transform Coefficients (TCOEF) and Motion Vector Difference (MVD).

For TCEF a table is provided giving the assigned codeword for 102 combinations of the set Run, Level and Last. If a combination is not listed in the table, then a special code will be assigned indicating that it is a special case, and a 13-bits code will follow describing the values.

In MVD the table is adapted for Motion Vectors coding. A table provides 65 codes associated to typical motion vectors values. Smaller motions will be coded with shorter words.

Levels and Profiles

For MPEG-4 levels and profiles can also be chosen to combine the tools required by the application. Four of the main profiles of MPEG-4 visual, Part2 are:

-Simple profile: supports Inter and Intra coded rectangular video object planes (VOP) for I and P VOPs respectively

-Advanced simple profile: this profile improves the compression efficiency of the simple profile, by supporting quarter-pixel resolution and global motion estimation in addition to B-VOPs. Alternate quantizer and interlaced video coding are used.

-Core profile: this profiles adds to the properties of the simple profile, P-VOP temporal scalability, B-VOPs coding and alternate quantizer but above all introduces video object coding with binary shape.

-Main profile: interlaced video and sprite coding are supported. Video object coding is carried out again but greyscale alpha maps are used instead of alpha planes.

Further information about object based encoding is given in next section.

VI. Object based Coding

One of the main innovations in MPEG-4 Visual is, without a doubt, the idea of replacing the conventional rectangular frame coding approach by an object-based one. A video sequence will be seen as a composition of video objects instead of a set of consecutive frames. [15]

In the Systems subsection of MPEG-4 Part 2 /Visual, the tools for object-based representation are specified.

VI.1 Video Objects Planes

The definition of a video object (VO), according to [23], is an area of the video scene that may occupy an arbitrary shaped region and may exist for an arbitrary length of time. If a particular instant in time is regarded, the VO will be denoted Video Object Plane (VOP) and will correspond to a layer of the frame for a determined object. If a sequence has one single VOP which size is fixed and equal to the size of the frame, the traditional frame-based point of view is regained.



Figure7 shows a frame composed of three video objects: VOP1: the sun, VOP2 the cloud and VOP3 the background.

For synthetic sequences and video sequences built using a chroma-key tool or any other composition method, the objects are already defined. Otherwise, a process of segmentation will have to be carried out to identify the different objects of the sequence. Several algorithms for video segmentation have been developed. Section VI.3 is exclusively dedicated to an overview of different methods of segmentation and their efficiency.

By either analysing the image composition data, either running the segmentation; each object is defined and its shape information obtained. This shape information can be described by a greyscale alpha plane or by a binary alpha plane. In the first case pixels are considered opaque (1) or transparent (0). In the second one, each

pixel's transparency is represented by 8 bits. In both cases the object has an arbitrary shape and the corresponding rectangular VOP has still to be formed. This should be done keeping in mind that we want to obtain the maximum coding efficiency, and with this intention, Ghanbari [9] suggests the following steps:

1. The tightest rectangle around the object must be generated. To avoid problems in case the sequence is 4:2:0 and so the dimensions of chrominance is half of the luminance, the top left position of the rectangle is an even numbered pixel and it will be the temporary origin pixel.
2. A control Macroblock is formed at the origin pixel defined in 1.
3. The origin point is slid to each even numbered pixel of the macroblock. From each new position of this origin point, to the right bottom side of the object a rectangle consisting in multiples macroblocks is generated. The number of macroblocks that contain at least one object pixel is computed for each new rectangle generated.
4. The numbers of significant macroblocks obtained in 3 are compared and the origin point which results on the smallest number is selected as the origin of the new rectangle.
5. By setting to zero the extended pixels, an extended boundary box is obtained: the VOP is formed.

If in step 1 the origin point is the origin of the frame, the formation procedure is skipped.

VI.2 Object-based video coding process

When the VOPs have been defined, three concurrent processing blocks can be differentiated when coding a video sequence: one for coding the VOP for the background, one for coding the VOP(s) for the foreground and one additional for coding the shape of this foreground.

VI.21 Shape Coding, alpha mask.

The decoder will need to identify the shape of each VOP, for this, alpha mask information has to be sent. The grey-scale alpha planes are encoded in a traditional way using motion compensation and DCT like it is usually done with texture. For a binary alpha plane instead, it is not so humdrum. The process, which is carried out per macroblock, is explained below. From now, the term macroblock MB will be used to define a 16 by 16 block.



Figure 8.

Figure 8. Transparent (1), Opaque (2) and Boundary macroblock (3).

For each macroblock of the VOP three situations can arise: all pixels are outside the object - their value is '0', all pixels are inside the object - their value is '1' - or finally there is a mixture of black and white pixels because the macroblock is a *boundary* macroblock. In the first two cases no further processing is carried out. Only the boundary macroblocks need to be coded. Several different methods can be used [9]: chain code, quad tree coding, modified modified reed,... In the last versions of the verification model, a Context based Arithmetic Encoding (CAE) is recommended as the best solution to code those Boundary Alpha Blocks (BAB).

CAE

In section II.32 an overview of arithmetic encoding has been presented. Context based arithmetic encoding is a binary arithmetic coding where the assigned probability is adapted according to the context of the neighbouring pixels.

A sequence of symbols X1 to X4 is to be encoded. Their probability of occurrence is provided. An initial range goes from 0 to 1 and, proportionally to their probability, the symbols are assigned some percentage within this range, dividing the range in sub-ranges. From the first symbol X1, the initial range is adjusted to its sub-range ([x11 x12] for example). The second symbol is analysed. Its sub-range ([x21 x22]) defined previously is rearranged proportionally to fit in the new range instead of the initial range 0 to 1. The new range when analysing symbol X3 is

$$[x_{11} + x_{21} * (x_{12} - x_{11}) \quad x_{11} + x_{22} * (x_{12} - x_{11})]$$

The same process is reiterated until all the symbols have been analysed. The last resulting range will be a really narrow one. Any point within this last range can be transmitted to provide the compressed information of the sequence of symbols.

CAE estimates the probability of a symbol using local and spatial information from the context symbols.

CA Encoding of the BABs.

Each boundary macroblock is treated separately after initializing the arithmetic encoder. The steps to follow are described below [23].

First, for each pixel X in the BAB, a context is computed. Depending on the type of frame we are facing, two different patterns for calculating the context can be chosen. For intra coded frames, only pixels from the current frame and spatially neighbouring pixel X are brought into play. For inter coded frames, the context is built from pixels both from the current BAB and the nearest forward motion compensated one. The motion operations carried out for alpha blocks differ in some aspects from the one explained in section III. These differences are described in next paragraph.

	C9	C8	C7	
C6	C5	C4	C3	C2
C1	C0	X		

Figure 9.

Template intra.

C3	C2	C1
C0	X	

	C8	
C7	C6	C5
	C4	

Templates inter.

In figure 9 the templates to obtain the context in each case are represented. For intra BABs the context comprises 10 bits. If any of the pixels is undefined, it will be given the value of the nearest neighbouring pixel of the BAB, unless it is above or to the left of the BAB, in which case they will be set to zero. For inter BABs the context is made of 9 bits; 5 of them come from a motion compensated BAB. The pixels can be scanned in raster or in vertical order, but the order chosen has to be signalled in the bitstream.

When the context has been calculated for a pixel X, a corresponding probability for binary arithmetic coding is read in one of the two tables included in MPEG-4 Visual Standard for CAE and attached in the CD annexed to this project. One corresponds to 10 bits, for intra coding and one to 9 bits for inter coding. The relevant probability $p(0)$ is obtained by dividing by 65535 the value in the table. The sub-range assigned to X for the CAE will be from 0 to $p(0)$ if the pixel X is '0' and from $p(0)$ to 1 if it is '1'.

For each macroblock after reshaping the range at each of the 256 pixels, we obtain a final narrow range. Any value within this range can be chosen to transmit the BAB's information.

This step of CA encoding is the last step that is specific to shape encoding.

Motion estimation and motion compensation for alpha planes.

The context calculated for inter BAB uses pixels resulting from motion estimation and compensation. The process of motion estimation and compensation is explained in section III Two main specific indications should be given to perform properly the compensation in case of dealing with an alpha mask. [9] If the region referred to is outside the VOP, the corresponding value is set to 0.

2. Only forward compensation is carried out. for P-VOPs the reference VOP is the previous I or P-VOP and for B-VOP the reference is the temporally nearest I or P-VOP. [23]

With the Context based Arithmetic encoding finishes the particular process of encoding the alpha mask. Zigzag scanning and VLC can be applied to the data previous to be multiplexed with the information from foreground and background coding.

VI.22 Foreground Coding.

The alpha plane will indicate to the decoder the shape of the objects. Those objects will be coded using a method similar to the traditional one explained in--- but with some extensions to fit the requirements of arbitrary shape objects. Two main steps can be recognised: Motion compensation and Texture coding.

Motion compensation of arbitrary shaped objects. Padding.

Motion estimation and compensation must be carried out for P-VOPs and B-VOPs. The systematic process of motion estimation and compensation has been explained earlier. However, this step is not to be underestimated as we are now dealing with arbitrary shaped areas and so a problem can arise: the reference macroblock can fall outside the VOP. To avoid this embarrassing situation a intermediate step has to be added which consists on padding the macroblocks of the reference VOP.

When referring to a macroblock in another frame, we can encounter two fussy situations: either the macroblock is totally transparent, either some of the pixels are transparent which the case for a boundary macroblock is. In both case a padding will have to be performed starting by padding all the boundaries macroblocks. [23][9]

Boundary Macroblocks padding.

Both horizontal and vertical extrapolation will be performed to pad the boundary macroblocks. First the non-transparent pixels of a row are extrapolated. If there are opaque pixels on both extremities of the row, the transparent pixels will be assigned a value equal to the mean between these two neighbouring pixels. If the row is only bordered on one side, the whole row is filled with the value of the nearest opaque pixel of this row to the transparent ones. After the horizontal padding is performed, a vertical padding of the resulting macroblock starts. The process used is the same applying the steps to columns instead of rows.

Exterior Macroblocks padding.

The macroblocks referred to can fall entirely outside the VOP. They will be totally transparent and will need to be padded as well. This can only be done when all boundary macroblocks have been treated as the padding for exterior macroblocks will be performed by replication of the neighbouring macroblocks's columns or rows. The choice of neighbouring macroblock is done according to figure 10. If MB_A is a boundary MB, then it will be used to proceed to padding. Otherwise MB_B will be selected if it is a boundary block. Next possible choice is MB_C and last MB_D. If no boundary MB is found in the neighbourhood, the non-transparent pixels are filled with the value 2^N with N the number of bits per pixel.

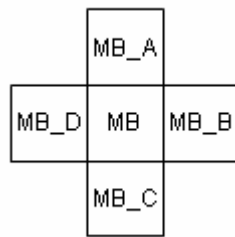


Figure 10.

Figure 10 Order for padding.

Ghanbari [9] places the emphasis on one point: The reference VOP is padded based on its shape information. When the reference VOP is smaller than the current VOP, the reference VOP is not padded up to the size of the current VOP.

A normal process of motion compensation can afterwards be performed. The motion vectors used for motion estimation are transmitted, and the difference between compensated image and current frame is sent after transformation.

Last step in video compression consists on spatial redundancy elimination by transformation.

DCT and Shape-adaptive DCT.

Next step in any compression algorithm is to encode the intra frame and the motion compensated inter frames.

In object-based coding, the opaque blocks of the VOPs will be coded with traditional 8 by 8 block DCT to each, luminance and chrominance. For Boundary blocks with both opaque and transparent pixels, two methods can be used: pad the blocks and run a DCT or apply a Shape Adaptive DCT (SA-DCT). In the first solution if we have a N by N block of pixels, we will have N by N coefficients. The shape adaptive DCT instead is more efficient because only opaque pixels are coded.

SA-DCT

The concept of SA-DCT is explained here. The VOP is processed per 8 by 8 blocks. For each block, each row is first treated. All the opaque pixels of the row are shifted to the left. The aligned pixels are then transformed by a one-dimensional DCT in the horizontal direction. A resulting block with coefficients is then treated column by column shifting all the non-zero coefficients to the top of the block. A second step of one-dimensional DCT is then carried out in the vertical dimension. [12] [24]

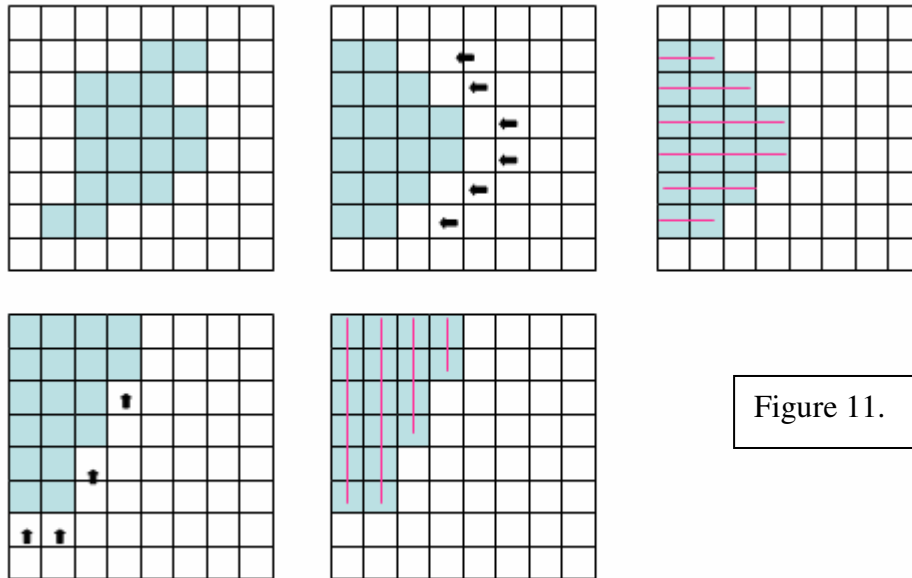


Figure 11.

Figure 11. SA-DCT. Original MB, horizontal shift, horizontal DCT, vertical shift and vertical DCT.

This process concerns the inter VOPs. For intra VOPs an additional step consists in computing first the zero-mean block. this variation of SA-DCT is called Δ SA-DCT.

This is the last step before zig-zag scanning, quantisation and VLC. In MPEG 4 Variable Uniform Quantization and Huffman based coding are used.

VI.23 Background coding.

The main advantage of object-based encoding is that it allows the background in a simpler way: first, it can be coded with lower reliability because it is usually a less important part of the image. the spectator is concentrated on the action involving the objects and do not pay so much attention to the background. Furthermore the background does not usually suffer much changes and so the efficiency in compression can be high. [17]

One method used is to send first a low quality approximation of the background and afterwards send progressively some areas of this background with higher quality.

Another common technique is based on transmission of a sprite panorama (still image describing a panorama of the background) once and, consecutively, for each frame a set of eight global motion parameters describing the motion.

MPEG-4 supports DWT coding of still images as seen in section V.3.

Those three steps - mask coding, foreground coding and background coding - compose the structure of object-based coding. By multiplexing the data and sending to

the user information of composition, or allowing it to interact with the objects, the goal of MPEG-4 to fit multimedia applications is fulfilled.

Object-based coding is not commonly used yet. First, the segmentation block still needs to be improved to allow better results. The Variable bit rate resulting from object-based encoding is another problem to take into account. The delays introduced cannot be ignored. Furthermore, an evaluation should be done before deciding if a sequence is prone to be coded with object based encoding. After all, it means sending one more block of information -shape information- and depending on the sequence the method could be useful or not.

However the idea is really promising, and by fixing some details and improving some deficiencies, the result can be more than satisfactory.

VI.3 Segmentation.

The first step when planning to perform an object based processing is to obtain those particular objects. Static image segmentation has already been widely investigated and several methods have been described ([7] [17] [9] [20]). The advantage when speaking about video sequence is that a tracking of the objects through consecutive frames allows a better detection.

Three main methods can be differentiated, but much more are used and the best option, without a doubt is to combine several of them.

Pixel based techniques.

Threshold techniques. They are based on local pixel information and consists usually in calculating a histogram and deciding a level or (levels) of threshold to divide the pixels into groups. Grey level but also colour, texture, etc can be the criterion.

Clustering techniques. Pixels are divided in different classes, using a certain algorithm i.e.: k-means. Each class of pixel is used to segment the image.

As no spatial information is taken into account, blurring boundaries are not well detected.

Region based techniques.

Growing. The principle is to group pixels into regions based on predefined criteria. These are the step to follow to perform growing:

- Compute at every pixel the same set of properties
- Choose an arbitrary seed pixel
- Compare it with neighbouring pixels
- Grow the region adding pixels that are similar for the property taken in account.
- Take another seed pixel when the area analyzed doesn't grow anymore.

This algorithm has some disadvantages: different choices of seeds may give different segmentation results and some problems can occur if the arbitrarily chosen seed point lies on an edge. The solution is to allow areas to grow simultaneously.

Splitting and merging. It consists in split the image in quadrants and merging adjacent quadrants or splitting them again if possible.

Edge based detection.

Local or global techniques can be used. One method consists in extracting edges by performing high-pass filtering using the Fourier operator. Some other differential operators are Roberts, Sobel, Prewitt and Laplacian operator. Gradient edge detectors are widespread.

None of those techniques is actually totally satisfactory yet and some more effort has to be made in improving segmentation.

VII. MATLAB implementation of an MPEG-4 style codec.

An MPEG-4 style codec in MATLAB has been implemented for this project. The main goal is to show the philosophy of an object based encoder by breaking up the structure into simple blocks. The result allows didactical approach, improvements and tests by only easily changing fragments of the code. The application does not produce a MPEG-4 compliant bit stream and can not be called Mpeg-4 encoder It is MPEG-4 *style*.

VII.1 Algorithm.

In figure 12 a schema shows the different blocks involved. Some more explanations come below.

First of all a segmentation has to be performed to detect the object from the static (or partially static background). From this function three layers can be distinguished: the background, the foreground and the mask.

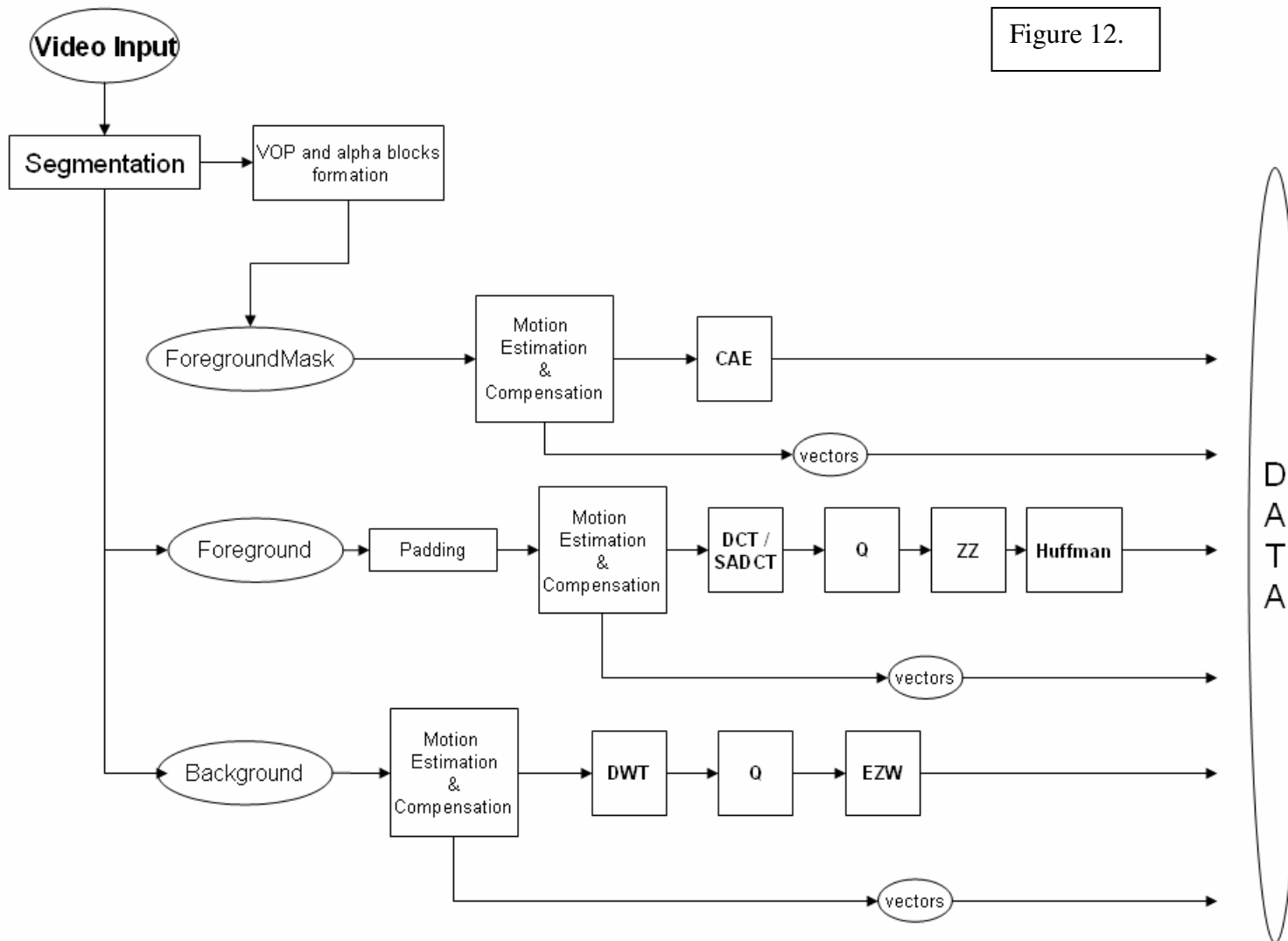
The mask will provide information about the shape of the object. This information is needed at the decoder to perform a reconstruction of the foreground respecting the contour. The first step to encode the mask is to proceed to a definition of the alpha plane: as explained before, the tightest rectangle around the object is first defined and modified afterwards to obtain the best boundary box which is the box that contains the whole object and at the same time the less number of macroblocks. The alpha plane is defined and can now be coded. Motion estimation and compensation are performed for P and B frames, and the I frames as well as the compensated P and B frames are Context arithmetic encoded.

The foreground is encoded in the classical way with some peculiarities to deal with the arbitrary shape of the object. At any moment, three types of macroblocks will be treated in a different way: the blocks outside the object are not coded, the blocks inside are coded in the traditional way, and the boundary blocks have to be analysed more carefully and a special processing will be applied. As usual, motion estimation and compensation are performed but to deal with reference areas pointed outside the VOP, a process of padding is needed. Then, a DCT block will transform the information into coefficients. For boundary blocks, the Shape adaptive DCT described in... as a more efficient one for blocks with transparent pixels is used. Quantization, zigzag ordering and Huffman coding are then performed in the traditional way.

For the background the DCT and its disadvantages are replaced by a DWT. The use of wavelet is another novelty of MPEG-4. An EZW algorithm is applied to entropy encode.

For the implementation of this MPEG-4 style encoder, some free source codes have been borrowed. More specifically, the segmentation, motion estimation and compensation, Huffman code and the background coding use available source code modified to fit the requirements of the present project; allowing the effort to be put on object-based distinctiveness of coding (VOPs definition, mask encoding, shape adaptive DCT).

Figure 12.



These source codes are available at MathCentral exchange of the webpage of MathWorks:

Segmentation: Available source code from the MATLAB Implementation of Graph-Based Foreground Segmentation of Nicholas R. Howe..

Motion Estimation and compensation: a folder contains all available source code from Aroh Barjatya. Several different algorithms can be chosen to carry out motion estimation (Exhaustive Search, Algorithm Three Step Search Algorithm, New Three Step Search Algorithm, Simple And Efficient Search Algorithm, Four Step Search Algorithm, Diamond Search Algorithm).

Huffman code: free source code from "Pepecito"

EZW: Ravi Lakkundi

VII.2 Code

A more detailed description of the code written is presented here. The whole code commented can be found in the attached CD.

Main.m

This is the main file to run the encoder. It reads an avi file, stores the frames and calls the function *extractForeground* to perform segmentation and obtain the mask layer. The sequence of frames of the three layers: mask, foreground, background are then defined and store in three different arrays of cells. Each of them is then coded by calling the corresponding function *Maskcoding*, *ForegroundCoding* and *BackgroundCoding*.

Maskcoding

This function calls *alphaBlock* which defines the alpha plane. *MotAlpha* is the used to perform motion estimation and compensation. For both P and B frames only one reference is taken. The vectors defining motion prediction are computed by any of the *motion estimator* functions available in the set of source code and the residual is obtained with the function *motionComp*. Finally the Context based arithmetic code is written in *CAEIblk* and *CAEPBblk* respectively for I and P or B frames with the correct pattern. The tables used for the probabilities in the CAE are provided in the cd.

ForegroundCoding

Every process has to be applied to luminance and chrominance samples. After defining the VOP, a set of padding steps is required (*function Padding*). First boundaries block are padded (*PadBound*) and afterwards external blocks suffer the process (*padExtBlkMot*). Those processes are carried out per Macroblock in order to apply motion estimation and compensation (*motForegr*). Here again the available source code is used, but first the reference frames have to be chosen and this time bilateral prediction is implemented. A pondered average between forward and backward predictor gives the vectors.

The compensated VOP pixels are then transformed. The *DCTransf* function divides the macroblocks into boundary or internal blocks to judge the type of DCT to apply: normal or shape-adaptive DCT (*SADCT* function). This SA-DCT is also different for inter or intra

frames. Finally quantization and zig-zag ordering are performed by *QuantZ*, *Qzz* and *zigzag*. Huffman coding is the last step implemented by *norm2huff*.

BackgroundCoding

Motion Estimation and compensation are once more performed with any of the *motion estimator* function and *motionComp*. Then wavelet transform and an EZW coding are performed by *ezw_encode*.

VII.3 Results

Some figure and tables are given below to illustrate some of the steps. First of all the decomposition unto VOPs is a fundamental starting point for the encoding process. Figures 13, 14 and 15 show respectively the Mask, Foreground and Background.

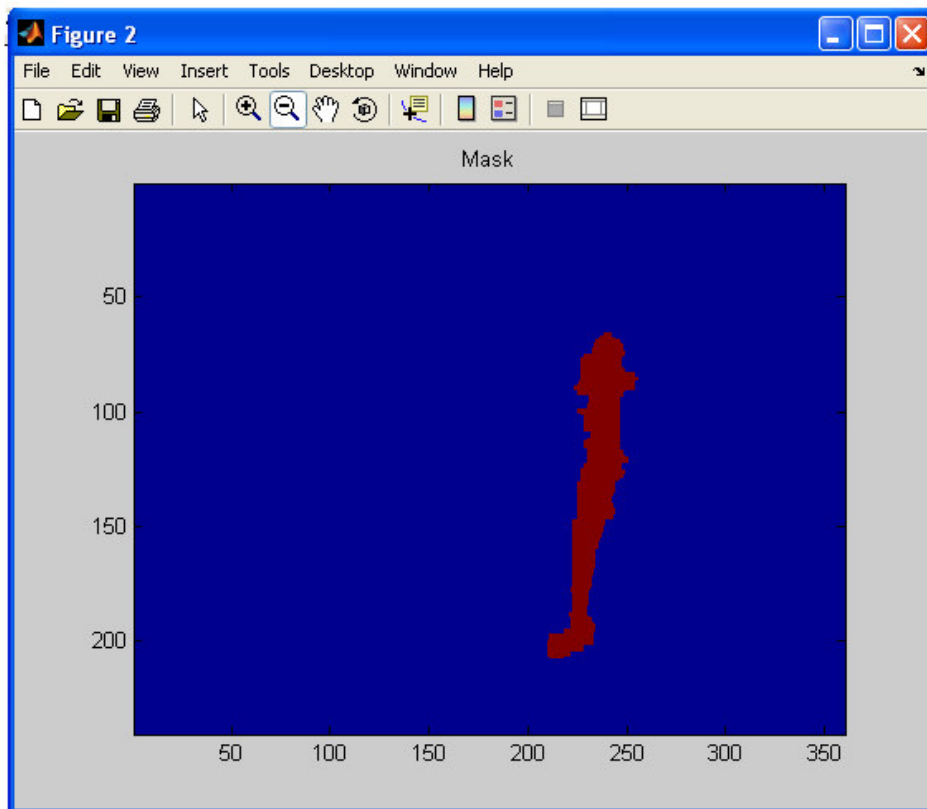


Figure 13.

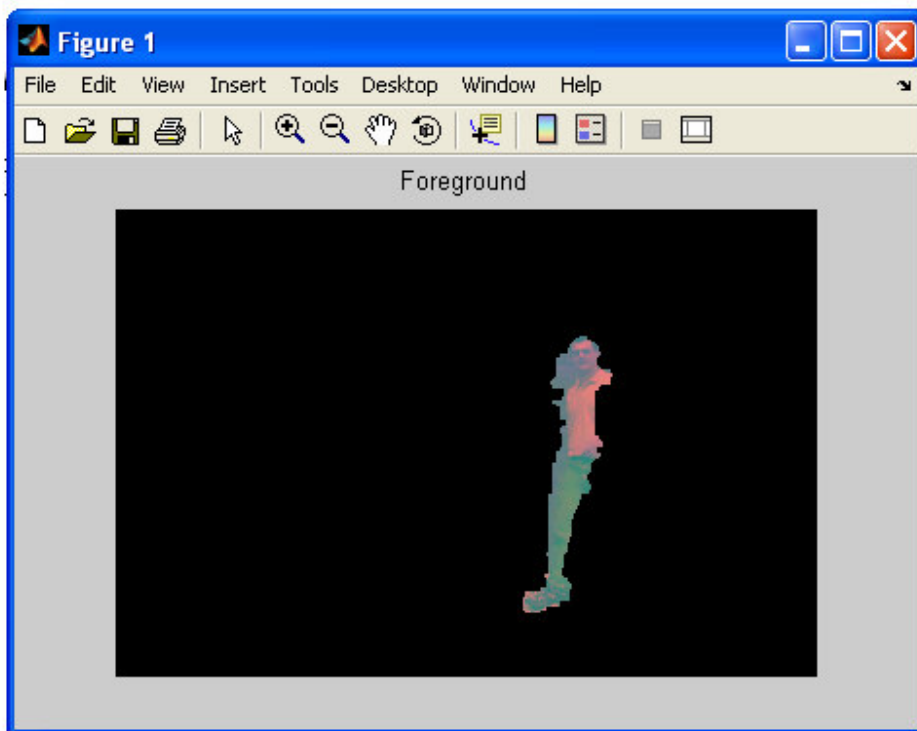


Figure 14.

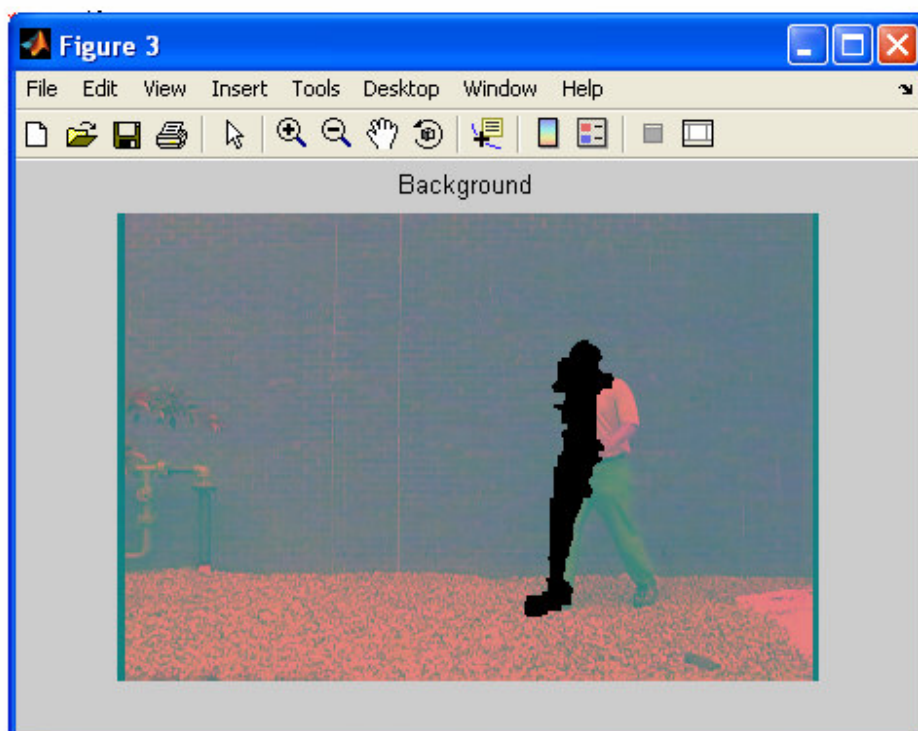


Figure 15.

When the mask is obtained, the process of defining the alpha plane starts. the result is shown in figure 16.

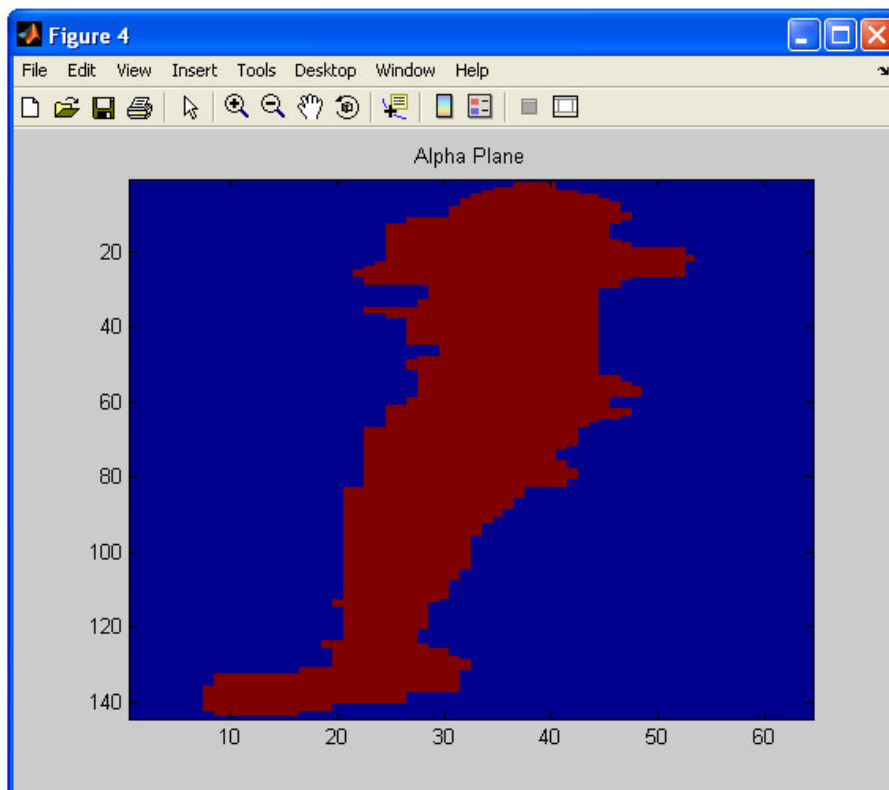
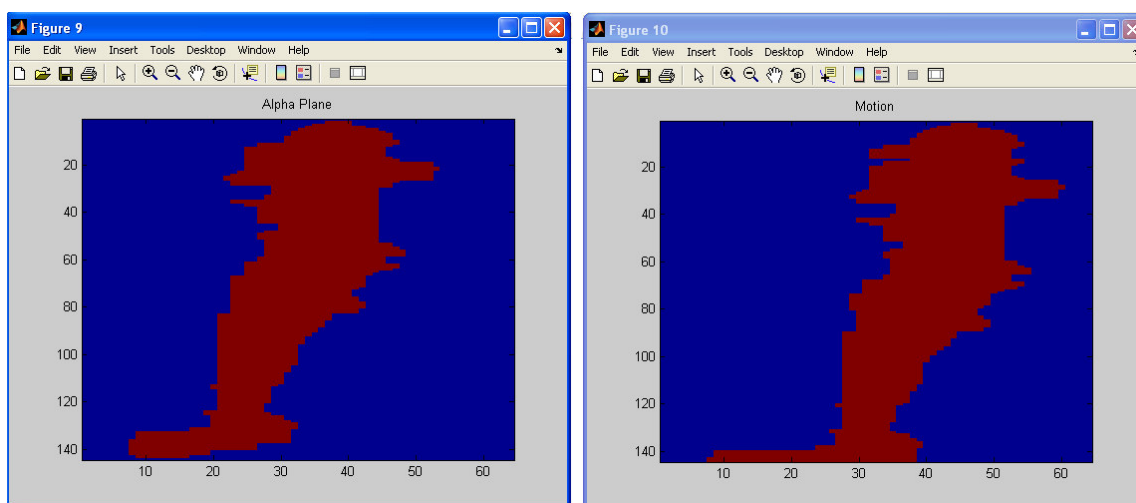


Figure 16.

We can see that the size of the Alpha plane is 144 by 64. This corresponds to 9 by 6 Macroblocks of size 16 by 16.

Figure 17.

The effect of Motion estimation can be noticed in figure 17.



Padding is another key step in object-based coding. This process effects are revealed in figure 18.

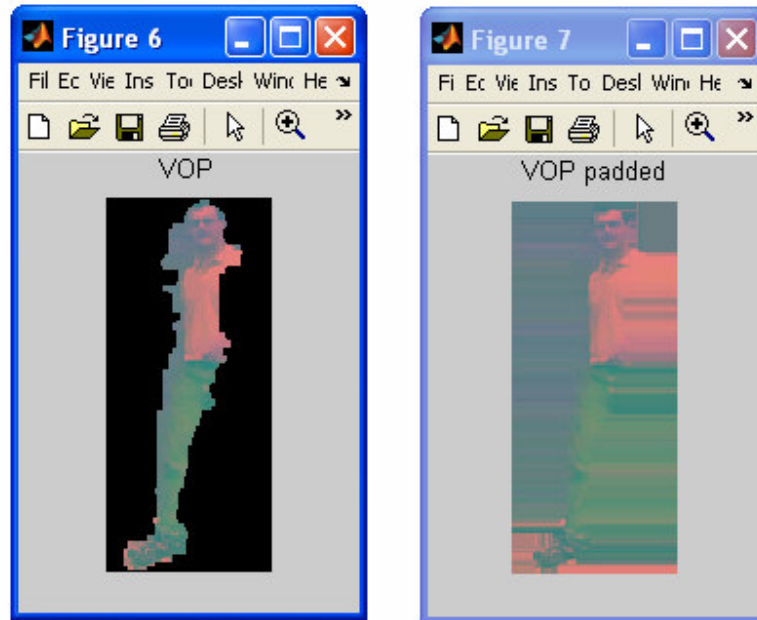


Figure 18.

The tables 5 and 6 show the Shape adaptive DCT performed on a 8 by 8 block. Table 5 is the boundary block before any type of processing and in table 6 the shifts and DCT in both directions have been implemented.

100	100	0	0	46	50	0	0
66	0	0	0	0	0	0	0
77	80	82	78	80	77	0	0
100	100	103	103	103	103	103	103
67	67	99	100	100	0	0	0
74	74	72	74	75	102	101	74
43	62	102	102	60	65	65	66
69	0	0	0	0	66	54	0

Table 5. Boundary block before DCT

52,326	17,869	0,75593	-8,8783	-0,86603	1,1154	0,46869	0,31218
72,578	24,637	1,0422	-12,128	-1,1315	1,4573	0,57403	0,38235
68,367	22,768	0,96318	-10,874	-0,86603	1,1154	0,33141	0,22075
61,529	19,757	0,83581	-8,8783	-0,46869	0,60363	0	0
52,326	15,756	0,66654	-6,2779	0	0	0	0
41,112	10,964	0,46384	-3,2497	0	0	0	0
28,319	5,6232	0,23788	0	0	0	0	0
14,437	0	0	0	0	0	0	0

Table 6: boundary block after Shape-adaptive DCT.

Finally the effects of the DWT can be observed in figure 19.

Figure 19.

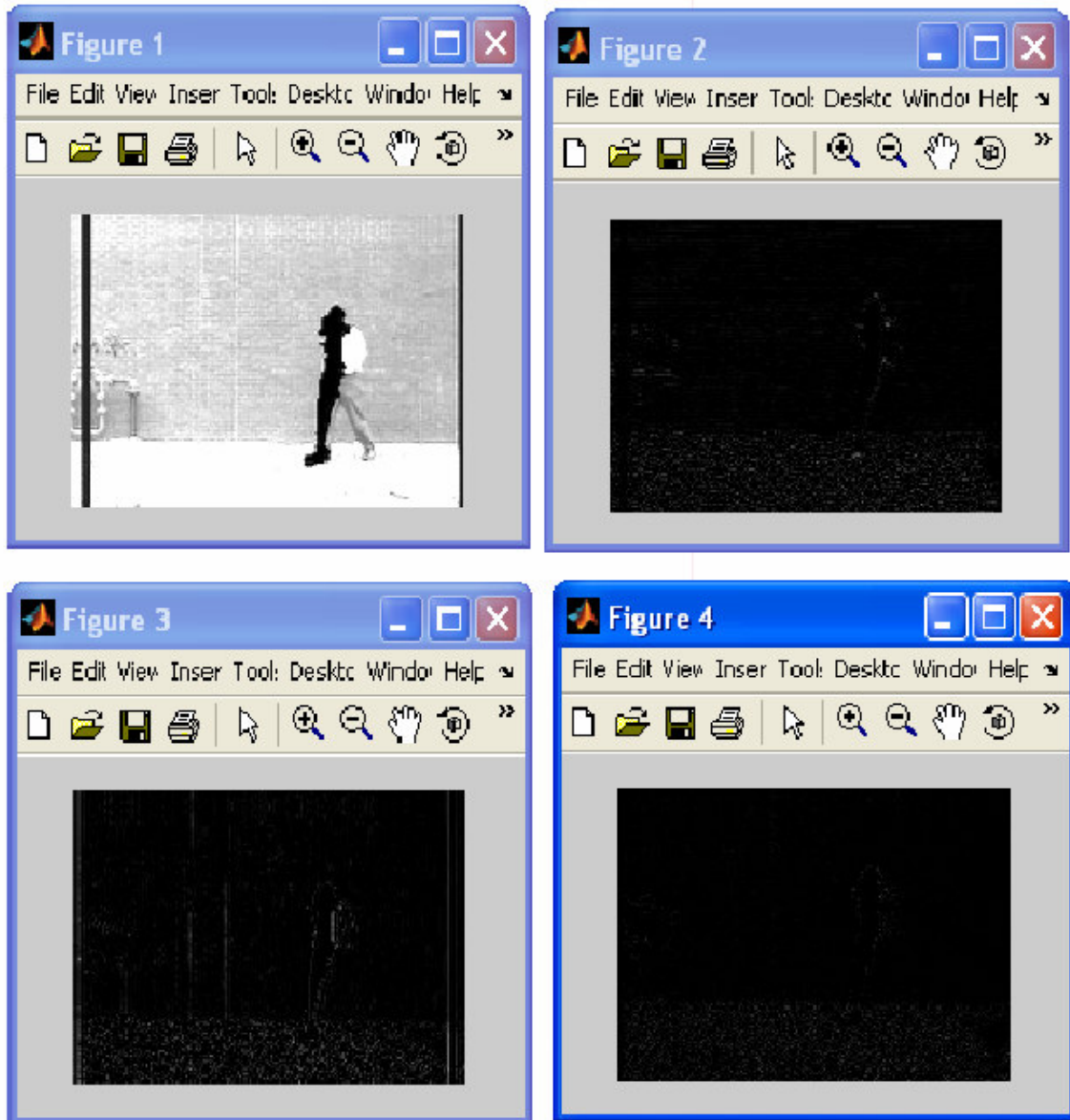


Figure 19. DWT applied to foregroundVOP

VII.4 Conclusions

The software implemented is just a didactical approach to MPEG-4. Several improvement can be made to increase the efficiency and optimize the time of computing which is definitely not viable. The memory requirements and the encoding delay are not to be sneered at. However it is a good starting point for understanding the philosophy of object-based encoding and because of the partitioned structure of the functions, it allows easy changes and tests.

The segmentation block can be replaced entirely as it is not a really accurate one. The motion estimation function allows playing with different search algorithms. The DWT can be improved and the idea of sending the background in a scalable way exploited. The tables for quantization are also easy to replace. The CAE uses raster order. A vertical order could be used and the differences analysed to evaluate the best solution. An alpha plane has been the base of the entire project. In the Main profile -as a difference with the core profile- a greyscale plane is used instead. This is another possibility of future work.

VIII. Conclusion

Object-based encoding presents promising results. An improvement of the coding efficiency, which is a fundamental aspect for application requiring transmission and storage, is expected. Furthermore, the possibilities that objects interaction offers is a key point when focusing in the user expectations and needs.

The profiles of the standard MPEG-4 are not yet commonly developed. Some problems need, indeed, to be solved before spread this technique. Variable bit rate, delays, segmentation deficiencies, memory requirements are some of the points where researchers should concentrate.

However the concept is definitely hopeful and the high speed technology is growing up let think that all profiles of MPEG-4 will soon invade every house.

References

- [1] Ayloo K., Encoding and decoding of MPEG-2 compressed video bitstream over a direct communication line, USA
- [2] Barjatya A., Block matching algorithms for motion estimation; 2004
- [3] Bhaskaran V., Konstantinides K., Image and video compression standards, USA; 1997
- [4] Bauer S., Kneip J., Mlasko T., Schmale B., Vollmer J., The MPEG-4 multimedia coding standard: algorithms, architectures and applications, Germany; 1999
- [5] Bretl W. and Fimoff M. MPEG2 tutorial; 1999
- [6] Bultheel A., Wavelets with applications in signal and image processing, 2003
- [7] Cheng H.D., Jiang X.H. Sun Y. Wang J., Color image segmentation: advances and prospects USA; 2000
- [8] Cheng L., Zarki M.E., The analysis of MPEG-4 core profile and its system design, USA
- [9] Ghanbari M., Standard codec- Image compression to advanced Video coding, UK; 2003
- [10] Hoelzer S., MPEG-2 overview and MATLAB codec project.
- [11] Karthik G.S.A. Encoding and decoding of MPEG-2 compressed video Bitstream over a direct communication line. USA.
- [12] Kaup A., Panis S., On the Performance of the Shape Adaptive DCT in Object-based coding of motion compensated difference Images, Germany; 1997
- [13] Keinert F., Wavelets and multiwavelets, USA; 2004
- [14] Lehane B., O'Connor N., Murphy N., MPEG-4 tools and applications: An overview, Irland, 2003
- [15] MPEGroup, Overview of the mpeg-4 standard, International organisation for standardisation Organisation internationale de normalisation iso/iec jtc1/sc29/wg11 coding of moving pictures and audio, 2002
- [16] Mitchell, J. L. MPEG video compression standard. USA, 2002
- [17] Morris T., Britch D., Object-based intra-frame wavelet video coding, United Kingdom; 2001

- [18] Nunes P., Pereira F., Object-based rate control for the MPEG-4 visual simple profile, Portugal; 1999
- [19] Pankaj N., Topiwala, Wavelet image and video compression USA; 1998
- [20] Petrov S., Image Segmentation With Maximum Cuts, University of California at Berkeley; 2005
- [21] Piron L., Kunt M. Differential Coding of alpha planes with adaptive quantization
- [22] Quinnell R.A., Introduction to MPEG-4 Video Compression
- [23] Richardson I. E. G., H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia, United Kingdom, 2003
- [24] Stasinski R., Konrad J., Reduced-complexity shape-adaptive dct for region-based image coding, USA; 1998
- [25] Schremmer, Multimedia Applications of the Wavelet Transform, Mannheim; 2001
- [26] Tran S.M., Preda M., Preteux F.J., Case study: a basic composing tool for editing system information MPEG-4, France; 2003
- [27] Villasenor J.D., Belzer B., Liao J., Wavelet filter evaluation for image compression, USA; 1995
- [28] Zhai F., Pappas T.N., Motion-compensated wavelet video coding using adaptive mode selection, USA
- [29] Valens, A Really Friendly Guide to Wavelets; 1999
- [30] Ya-Qin Z. Motion-Compensated Wavelet Transform Coding for Color Video Compression, Transactions on circuits and systems for video technology; 1992

Content of the CD

In the cd attached, the source code for this software can be found in folder 'MPEG4MatLab'. This folder includes a video sequence to test and the tables for CAE. By simply running the Main function, the process of encoding will start. As a result three tables of data, one for foreground, one for background and one for mask will be obtained.

A pdf version of this report is also included.