Codification of Multumedia signals

Project documentation

Video coding

Lisbon, 2015

Tartalomjegyzék

[1 Introduction 3](#_Toc423983139)

[1.1 Compression of I-frames 3](#_Toc423983140)

[2 The objectives of the project 5](#_Toc423983141)

[2.1 Part 1 – Working with only I-frames 5](#_Toc423983142)

[2.2 Part 2 – Using I and P-frames 5](#_Toc423983143)

[2.3 Part 3 – Motion estimation 6](#_Toc423983144)

[2.3.1 The motion estimation process 7](#_Toc423983145)

[2.3.2 The motion compensating method 8](#_Toc423983146)

[3 The results 9](#_Toc423983147)

[3.1 The energy of the frames 9](#_Toc423983148)

[3.2 The entropy of the frames 10](#_Toc423983149)

[3.3 The achieved compression ratios 10](#_Toc423983150)

[3.4 The signal-to-noise ratios 11](#_Toc423983151)

[4 Conclusion 12](#_Toc423983152)

# Introduction

Several video coding standard – including MPEG-4 and its successors – use the concept of I- and P-frames. I frames mean the reference frames, used to compress the following frames in the video more efficiently. P-frames are the frames, using the I frames for their compression. I-frames can be used for subtraction, simply coding the difference between them and the P-frames, or a more complex method, the motion estimation can be used.

## Compression of I-frames

The compression of the I-frame is simply done according to the JPEG standard. The original frame is compressed, and the compressed frame is sent to the decoder for decompression.

JPEG s a commonly used method of lossy compression for digital images. The term "JPEG" is an acronym for the Joint Photographic Experts Group, which created the standard.

The JPEG compression method uses the concept of macroblocks. Macroblocks are 8x8 blocks of the image. So the encoding process starts with dividing the image into macroblocks. After this, the Discrete Cosine Transform (DCT) is applied to each macroblock. In the standard, the fact that the human vision system is more sensitive to the changes of luminance values than to the chroma values is exploited. So instead of the RGB color space, the YCrCb colos space is used, and the chroma calues are usually subsampled. The DCT transform results in frequency values. In macroblocks, the high frequency values are usually very small, because nearby pixels don’t often change values drastically. This can result in zeroing a lot of high frequency values. And these zeroes can be exploited, resulting in a more efficient compression ratio.

After the DCT transform, a quantization is applied, its intensity depends on a quality factor parameter. This is the most lossy process of the JPEG compression.

The third main part of the compression is the coding. The DC and AC components, resulting from the DCT transform are coded in different ways. The DPCM (different pulse-code modulation) method is used for the coding of the DC values of the blocks. That basically means the following: the DC component of the first block is coded regularly, but for the following blocks, the previous DC values is subtracted from them, and only the difference is coded. Then we make symbols for these values, storing a size and an amplitude value. The size shows how many bits are needed to code the DPCM value, and the amplitude value is the one’s complement value of the DPCM value. The size value is Huffman-coded in the bitstream, the amplitude is pure binary.

In case of coding the AC values, we exploit that among the values of a block there usually are a lot of zeroes, as the frequency increases. So we use a so-called runlength coding. The coding results a set of two symbols. The first symbol contains the runlength and the size, the second is the amplitude. Runlength means that how many zero value there are between two non-zero values. The amplitude is the one’s complement of the value following the zero-series. Size is the same as it was in case of the DC components. Symbol one, that is the runlength and the size are Huffman-coded, while the amplitude is again a pure binary value int he bitstream.

# The objectives of the project

Our project was to investigate different methods, used for video coding, implement them, then check their efficiency with several measurements.

We investigated three different methods. In the first, we only used I-frames, that means that each original frame is compressed according to the JPEG standard, no similarity between the frames are utilised.

In the second one, we started to use P-frames, too, so we utilised the similarity between a main frame (the I-frame) and the following ones. We only compressed the difference between the I-frame and the different P-frames, according to the JPEG standard. (In case of the I-frame, the original frame is compressed, of course.)

The third case meant the introduction of motion estimation/compensation. This method is described in detail later.

## Part 1 – Working with only I-frames

This first part of the project was simple, it didn’t exploited the similarities between the frames of the video following each other. It is the same as compressing a bunch of pictures independent from each other.

The language used for the implementation was Python. The parts of this excercise was reading in the frames from a directory, compress them using Python’s standard JPEG compression methods, then writing them to files. After this, the measurements could be done, which we used for comparing the three different video coding method. These measurements are detailed in the results section.

## Part 2 – Using I and P-frames

In this part of the project, the fact that the pictures to be coded are part of a video, so they are similar were exploited. The first picture was the reference frame, i.e. the I-frame, the following pictures were P-frames. Figure 1 shows the block diagram of what this compression method does.



Figure 1 - Block diagram of dealing with the P-frames

First, we have to compress the I frame according to the JPEG standard. This JPEG is read back then, for the compression of the P-frames. It is important to use this version of the I-frame, not the original one, because the decoder side will only have this compressed version of the frame. (The left side of the figure illustrates the encoder, right from the dashed vertical line is the decoder.)

The rederence I-frame is subtracted from each P-frame, then this difference picture is the one compressed. The decoder side gets these difference pictures, adding them to the I-frame, then we have the reconstructed frames.

## Part 3 – Motion estimation

The most complicated, but also most effective method investigated during this project for video coding is the use of motion estimation/compensation between frames. Figure 2 shows the block diagram of this part of the project.



Figure 2 - Block diagram of the Motion Estimation method

In this case, the compressed, read back reference I-frame is not simply subtracted from the actual frame, but a more complex motion estimation process is applied. This process results in a set of motion vectors and a new image. This new image is then subtracted from the actual P-frame, the resulted image is the one which becomes compressed.

The decoder gets th eset of motion vectors and the compressed image, according to the P-frame, and with a motion compensation process, it reconstructs the P-frame. (In case of the I-frame, the same, simple JPEG compression is applied.)

### The motion estimation process

This process gets the I-frame and the actual P-frame as inputs. We split the P-frames to 16x16 blocks. The aim is finding the most similar block in the reference frame, according to each block in the actual frame.

For this, several methods can be used. We used a simple, quite resource-demanding, but effective one. A larger window is assigned to each block. In the part of the reference frame, defined by this window, we perform a full search. This means we iterate through 16x16 blocks, alternating the x coordinate of the top left pixel only by one each time, then after a row, alternating the second coordinate by one pixel.



Figure 3 - The motion estimation process

The similarity of the blocks is calculated in the following manner. We form the difference of the reference and the actual blocks, take the absolute value of this difference matrix, then summarize its elements. The block with the least value will be the most similar to the actual frame.

Into the actual block’s position of the image returned by this method, we put the most similar block from the reference I-frame. The returned motion vector belonging to the actual block is made by subtracting the coordinates of the actual block from the coordinates of the most similar block in the I-frame.

### The motion compensating method

The decoder uses this method to reconstruct the P-frames. Its two inputs are the reference image and the motion vectors according to the actual P-frame. It iterates through the blocks of the intermediate image to reconstruct, and putting the blocks from the reference image there, according to the motion vectors.

After this, the decompressed image belonging to the actual P-frame is added to the intermediate image returned by the motion compensating method. Then, we have our reconstructed P-frame.

# The results

For the comparison of the three different method, we used the following measurements. We measured energy and entropy values, signal-to-noise ratios between the original and the restored frames, and compression ratios. The results can be seen in the following figures.

## The energy of the frames

As it can be seen, the energy of the compressed frames is a little bit smaller, but very similar between each frame.

With the use of P-frames, the energy decreases drastically for the P-frames.

We got similar energy values in case of the motion estimation method as with the subtractive method.

## The entropy of the frames

The entropy values for the P-frames decreases, and decreases even more with the introduction of the motion estimation.

## The achieved compression ratios

We got significantly better compression ratios for the P-frames with the subtractive method, and it can be enhanced even more with the motion estimation.

## The signal-to-noise ratios

From the below figure it can be seen that the signal-to-noise ratio does not change significantly with the introduction of the more complex compression methods.

# Conclusion

During this project, we implemented three different methods for coding frames of a video segment. From the measurements we made it clearly can be seen that with the complexity of the methods, the better results can be achieved. The introduction of P-frames significantly increased the compression ratio, decreased the entropy, while keeping the signal-to-noise ratio similar to the method when we considered each frame as I-frames. The motion estimation method is much more complex than the previous two, but it can further enhance the results.