VIDEO COMPRESSION

USE OF VIDEO COMPRESSION

A simple calculation shows that an uncompressed video produces an enormous amount of data: a resolution of 720x576 pixels (PAL), with a refresh rate of 25 fps and 8-bit colour depth, would require the bandwidth of around 1.7Mb/s. And if the resolution is HDTV then it can go up to 1.6Gb/s which is nearly impossible task .So we compress video so that less bandwidth is required.

MPEG COMPRESSION

The MPEG compression algorithm compress the data in 5 different steps.

STEP1: Resolution Reduction

The human eye has a lower sensibility to colour information than to dark-bright contrasts. A conversion from RGB-colour-space into YUV colour components help to use this effect for compression. The chrominance components U and V can be reduced (subsampling) to half of the pixels in horizontal direction (4:2:2), or a half of the pixels in both the horizontal and vertical (4:2:0). The subsampling reduces the data volume by 50% for the 4:2:0 and by 33% for the 4:2:2 subsampling:

STEP2: Motion Estimation

It uses three types of frames: I-frames (intra), P-frames (predicted) and B-frames (bidirectional).
The I-frames are “key-frames”, which have no reference to other frames and their compression is not that high. The P-frames can be predicted from an earlier I-frame or P-frame. P-frames cannot be reconstructed without their referencing frame, but they need less space than the I-frames, because only the differences are stored. The B-frames are a two directional version of the P-frame, referring to both directions (one forward frame and one backward frame). B-frames cannot be referenced by other P- or Bframes, because they are interpolated from forward and backward frames. P-frames and B-frames are called inter coded frames, whereas I-frames are known as intra coded frames.
The references between the different types of frames are realised by a process called motion estimation or motion compensation. The correlation between two frames in terms of motion is represented by a motion vector. The resulting frame correlation, and therefore the pixel arithmetic difference, strongly depends on how good the motion estimation algorithm is implemented. Good estimation results in higher compression ratios and better quality of the coded video sequence. However, motion estimation is a computational intensive operation, which is often not well suited for real time applications.following are the steps for motion estimation

Frame Segmentation - The Actual frame is divided into nonoverlapping blocks (macro blocks) usually 8x8 or 16x16 pixels. The smaller the block sizes are chosen, the more vectors need to be calculated; the block size therefore is a critical factor in terms of time performance, but also in terms of quality: if the blocks are too large, the motion matching is most likely less correlated. If the blocks are too small, it is probably, that the algorithm will try to match noise. MPEG uses usually block sizes of 16x16 pixels.

Search Threshold - In order to minimise the number of expensive motion estimation calculations, they are only calculated if the difference between two blocks at the same position is higher than a threshold, otherwise the whole block is transmitted.

Block Matching - In general block matching tries, to “stitch together” an actual predicted frame by using snippets (blocks) from previous frames. The process of block matching is the most time consuming one during encoding. In order to find a matching block, each block of the current frame is compared with a past frame within a search area. Only the luminance information is used to compare the blocks, but obviously the colour information will be included in the encoding. The search area is a critical factor for the quality of the matching. It is more likely that the algorithm finds a matching block, if it searches a larger area. Obviously the number of search operations increases quadratically, when extending the search area. Therefore too large search areas slow down the encoding process dramatically. To reduce these problems often rectangular search areas are used, which take into account, that horizontal movements are more likely than vertical ones.

Prediction Error Coding - Video motions are often more complex, and a simple “shifting in 2D” is not a perfectly suitable description of the motion in the actual scene, causing so called prediction errors. The MPEG stream contains a matrix for compensating this error. After prediction the, the predicted and the original frame are compared, and their differences are coded. Obviously less data is needed to store only the differences.

Vector Coding - After determining the motion vectors and evaluating the correction, these can be compressed. Large parts of MPEG videos consist of B- and P-frames as seen before, and most of them have mainly stored motion vectors. Therefore an efficient compression of motion vector data, which has usually high correlation.

Block coding – Using Discrete cosine transform(DCT).

DCT

DCT allows, similar to the Fast Fourier Transform (FFT), a representation of image data in terms of frequency components. So the frame-blocks (8x8 or 16x16 pixels) can be represented as frequency components.It is used in audio compression as well .After motion compensation or motion estimation we have found vectors relating every frame so we only now have to use Discrete cosine transform to compress the frames.The DCT is unfortunately computational very expensive and its complexity increases disproportionately (O(N^2)). That is the reason why images compressed using DCT are divided into blocks. Another disadvantage of DCT is its inability to decompose a broad signal into high and low frequencies at the same time. Therefore the use of small blocks allows a description of high frequencies with less cosineterms.

QUANTIZATION

During quantization, which is the primary source of data loss, the DCT terms are divided by a quantization matrix, which takes into account human visual perception. The human eyes are more reactive to low frequencies than to high ones. Higher frequencies end up with a zero entry after quantization and the domain was reduced significantly.If the compression is too high, which means there are more zeros after quantization, artefacts are visible . This happens because the blocks are compressed individually with no correlation to each other. When dealing with video, this effect is even more visible, as the blocks are changing (over time) individually in the worst case.

ENTROPY CODING

The entropy coding takes two steps: Run Length Encoding (RLE) and Huffman coding . These are well known lossless compression methods, which can compress data, depending on its redundancy, by an additional factor of 3 to 4.