Mesh Based Interpolative Coding (MBIC)

Eckhart Baum, Joachim Speidel
Institut für Nachrichtenübertragung, University of Stuttgart

An alternative method to H.263 encoding of moving images at bit rates below 64 kbit/s is presented using adaptive spatial subsampling, motion estimation and interpolation. The non-equidistant sampling points (nodes) for luminance and colour difference signals are linearly interpolated, forming a triangular mesh which fulfills the Delaunay criterion. For a given number of nodes which corresponds approximately to the total bit rate, the mean square error between the original picture and the triangular mesh is minimized. The samples are entropy encoded using the Arithmetic Coding technique. For moving images, each node is associated with a motion vector. Simulation results of a complete encoder and decoder show that this method can provide lower bit rates than conventional schemes at a given picture quality for sequences with moderate movement.

1 Introduction

Image coding is required to transmit or store video sequences efficiently. In the past years transform coding schemes have been investigated and standardized for still images, e. g. JPEG coder. For moving pictures, hybrid coding schemes turned out to be very efficient. With these methods, the transform coding principle is applied to motion compensated frame differences. Several existing coding standards, MPEG-1, MPEG-2, H.261 and H.263 belong to this category.

All these methods use regularly sampled pictures as input signals where the spatial sampling points are equidistant, both in horizontal and vertical direction. In natural pictures, however, many regions show a flat distribution of the luminance and the colour difference signals. This also holds for (motion compensated) frame differences. In these areas, many samples are similar and thus highly correlated. With present hybrid coders, the redundancy is reduced by using the energy compaction property of the transform and by subsequent coarse quantization of the transform coefficients.

In this paper a method of adaptive, non-equidistant sampling is investigated. In flat areas, the number of sampling points is drastically reduced and only a few samples are coded. This provides a significant reduction of bit rate. To prevent from spectral aliasing effects due to subsampling, the sampling points are chosen as to minimize a local mean square error (MSE) between the original and the interpolated picture. For interpolation, the samples are connected by lines in such a way that a Delaunay mesh, well known from mathematical topology, results. This is why we call this new coding algorithm *Mesh Based Interpolative Coding* (MBIC).

In contrast to other interpolative techniques [1] the proposed algorithm MBIC uses a two-dimensional interpolation. The sampling points are automatically placed by the algorithm depending on the image content. Inter frames are handled by moving the points from frame to frame and coding the corresponding vectors. The decoder reproduces the missing pels by interpolating the received sampling points.

The paper is organized as follows: Section 2 describes the functionality of MBIC for intra coding of still images in detail. Section 3 deals with the extension to inter frame coding. Simulation results are presented in Section 4.

2 Intra Coding

Intra coding is applied to still images as well as to the first picture of a moving video sequence.

The proposed method reduces image data by leaving out most of the pels and transmitting only a small number of sampling points. In order to obtain very low bitrates, MBIC is a lossy coding technique. Therefore the interpolated picture nor-

8. Dortmunder Fernsehseminar; Sept. 1999; Multimedia: Anwendungen, Technologie, Systeme; ITG-Fachbericht 156; VDE Verlag 1999; S. 241-246

Fig. 1: Selected sampling points for the first frame of *claire* sequence

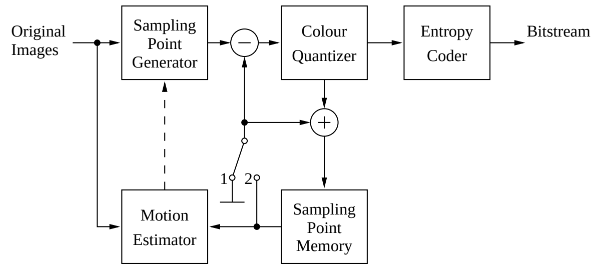

Fig. 2: Principle block diagram of the MBIC coder

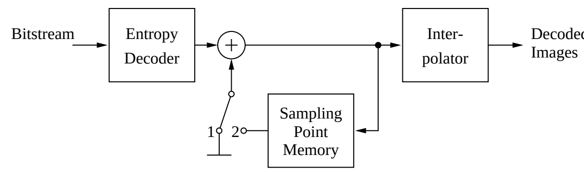

Fig. 3: Principle block diagram of the MBIC decoder

mally differs from the original picture and minor distortions have to be accepted. If the image contains many details between the sampling points, the error due to interpolation increases. Obviously the image quality is improved in this case by a denser set of sampling points. On the other hand lots of transmitted samples cause a high bit rate. For that reason MBIC uses an *adaptive* sampling. This means the sampling points are not equally distributed, and their density depends on the image content. In regions with many details the density is high, whereas a low density is preferred in regions with only few details. As an example **Fig. 1** illustrates an image of *claire* sequence and the automatically selected sampling points are indicated in the lower part.

Fig. 2 and **Fig. 3** show the principle block diagrams of the coder and decoder. The switches are used to select intra mode (position 1) or inter mode (position 2). The sampling point generator at the coder determines position and colour for each point. At the decoder side the corresponding block is the interpolator. Both sampling point generation and interpolation should not be considered independently, because the interpolation is also needed within the generation process of the sampling points.

In the proposed scheme of Fig. 2 and Fig. 3 a two-dimensional linear interpolation is used, as its implementation is much less expensive compared to higher order interpolation methods. Thus the sampling points can be placed by an iterative procedure, which requires frequent recalculations of the interpolated pixels. In the following, the main components of the coder are described in more detail.

2.1 Interpolation

In order to apply a two-dimensional linear interpolation to a set of non-equidistant sampling points, the algorithm generates a 2-D triangular mesh in a first step, as shown in **Fig. 4**. The mesh consists of n_t non-overlapping triangles, n_e edges and n_n nodes. It is fully connected, i. e., there is no pair of nodes, which could be connected additionally without crossing an existing edge. Each sampling point forms one node N_i ($i \in \{1, \dots, n_n\}$) of the mesh. The sampling points and consequently the nodes may have arbitrary positions and colours. Therefore N_i must be described by two vectors, the position $p_i = (x_i, y_i)$ with the two coordinates x_i and y_i, and the colour $c_i = (Y_i, Cb_i, Cr_i)$ with luminance Y_i and chrominance components Cb_i and Cr_i ($i \in \{1, \dots, n_n\}$).

An example of a triangular mesh is shown in Fig. 4. N_1, N_2 and N_3 denote three nodes with positions $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, $p_3 = (x_3, y_3)$ and colours c_1, c_2, c_3 respectively. Each pel of the picture, that does not lie on an edge of the mesh, belongs to exactly one triangle. Its interpolated colour depends only on the colours of the three surrounding nodes.

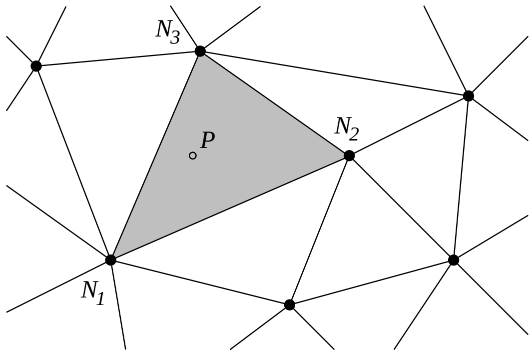

Fig. 4: Triangular mesh

In Fig. 4 the marked pel P with coordinates x, y belongs to the grey shaded triangle. Geometrically viewed, the three luminance components Y_1, Y_2 and Y_3 of N_1, N_2 and N_3 span a plane in the three-dimensional space $Y(x, y)$. Accordingly, two other planes are spanned by Cb_i and Cr_i, respectively. The three planes determine the colour $c(x, y) = (Y(x, y), Cb(x, y), Cr(x, y))$ of P by the following equations:

$$c(x,y) = \lambda_1(x,y)c_1 + \lambda_2(x,y)c_2 + \lambda_3(x,y)c_3 \quad (1)$$

with

$$\lambda_1(x,y) = \frac{y_2 - y_3}{\Delta}x + \frac{x_3 - x_2}{\Delta}y + \frac{x_2 y_3 - x_3 y_2}{\Delta} \quad (2)$$

$$\lambda_2(x,y) = \frac{y_3 - y_1}{\Delta}x + \frac{x_1 - x_3}{\Delta}y + \frac{x_3 y_1 - x_1 y_3}{\Delta} \quad (3)$$

$$\lambda_3(x,y) = \frac{y_1 - y_2}{\Delta}x + \frac{x_2 - x_1}{\Delta}y + \frac{x_1 y_2 - x_2 y_1}{\Delta} \quad (4)$$

and

$$\Delta = x_1 y_2 + x_2 y_3 + x_3 y_1 - x_1 y_3 - x_2 y_1 - x_3 y_2 \quad (5)$$

A pel that lies exactly on an edge connecting nodes N_i and N_j, belongs to both triangles which have this edge in common. Therefore assigning the pel to a triangle is ambiguous in this case. However, it does not matter which triangle to use for interpolation, because the interpolated value only depends on N_i and N_j, but not on the third node that would be different for both triangles.

As the interpolation procedure is implemented in both coder and decoder, they should produce the same interpolated image for a given set of nodes. This requires the same interplation rules Eq. (1) – (5) in the coder and decoder. Furthermore, as the interpolated values depend on the assignment of the pels to the triangles, the topology of the mesh plays an important role. To clarify this, **Fig. 5** illustrates two different meshes based on the same set of nodes. The pels at nodes N_1 and N_3 are assumed to be black, while the pels at nodes N_2, N_4 and N_5 are assumed to be white. Obviously, there are several possibilities to triangulate a given set of nodes. This implicates differences between the resulting pictures.

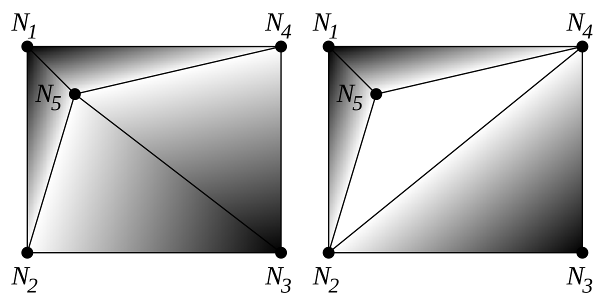

Fig. 5: Two possible triangulations

In order to obtain exactly the same interpolated images in the coder and decoder without transmitting additional information about the topology of the mesh, a unique triangulation to a given set of nodes is required. Moreover, with regard to the node generation described below, the cost of triangulating the nodes should be as small as possible. In particular, moving a node must not implicate a re-triangulation of the whole mesh, only few surrounding triangles should change. Both requirements are fulfilled by the *Delaunay triangulation* [2], which is therefore used by MBIC. Delaunay triangulations have also been used in [3], [4] for motion compensation as an alternative to block matching.

2.2 Node Generation

The generation of the nodes (or sampling points) is an important part of the coder, because the spatial distribution of the nodes has strong influence on the image quality, as explained above. In order to find convenient positions for all nodes, MBIC performs an iterative procedure based on node movement, deletion and insertion.

The algorithm starts with a regular mesh as shown in **Fig. 6** (1). First, the position of each node is locally optimized by moving it up to $\pm a$ pixel in any direction, in order to decrease the mean square error (MSE) between the surrounding triangles and the original picture. $a = 1$ has shown to be effective [5]. After that, nodes are removed, if their removal does not result in a MSE of the changed triangles which exceeds an upper limit. Finally, additional nodes are inserted into triangles, so that the MSE is reduced furthermore. The allowed node positions are restricted to the pel raster. Fig. 6 (2) illustrates an intermediate result. The same steps are carried out sequentially as shown in Fig. 6 (3) – (6). The implemented algorithm always converges to a state, where no more changes are to be applied. At this point the algorithm ends.

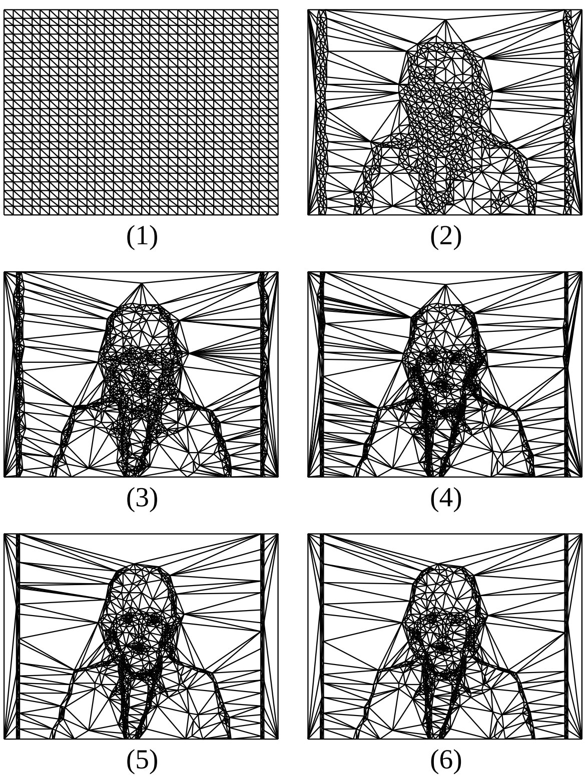

Fig. 6: Iterative mesh generation

The colours of the nodes are handled separately. At the beginning, each node is set to the colour of the corresponding pel within the original image. Finally, all colours are globally optimized with respect to the overall MSE.

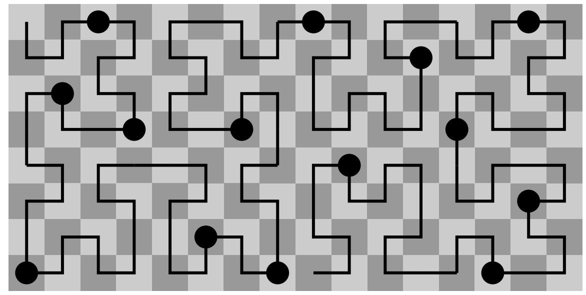

Fig. 7: Hilbert scan, grey shaded areas indicate pel positions, dots indicate sampling points

2.3 Node Parameter Coding

In order to encode positions and colours of the nodes, at first the nodes are arranged in a sequential order. This is done by scanning all pel positions along a *Hilbert curve* [6], [5]. As an example, **Fig. 7** shows a path through a 16×8 pel clipping of the image. Using this scheme, the order of the nodes marked by the black dots is determined such that successive nodes lie closely together, i. e. their positions and colours are correlated.

To take advantage of this correlation, the colours of successive nodes are differentially encoded and quantized. Instead of coding x and y coordinates, the position of a node is described by its distance from the preceding node along the Hilbert curve. Finally, an arithmetic coding reduces redundancy of position and colour information.

3 Inter Coding

Inter coding is applied to image sequences, in order to benefit from the correlation of successive frames. As described, a picture is represented by a number of nodes with parameters “position” and “colour”. For moving images, both node parameters may change from frame to frame. It is sufficient to transmit only these changes, if the decoder applies them to the parameters of the preceding image and creates the new mesh with it. By changing the node parameters, both movement and colour modifications can be realized. Hence there is no need to transmit a difference picture.

Main emphasis has to be put on the node generation algorithm of the proposed scheme. If the nodes of two successive frames are generated in-

Sequence	H.263		SPIHT		MBIC	
	bit/picture	PSNR (dB)	bit/picture	PSNR (dB)	bit/picture	PSNR (dB)
akiyo	15120	35.2	15460	35.8	10418	35.8
carphone	19024	35.0	17487	35.3	14070	35.4
claire	14736	38.1	14446	38.3	9158	38.4
container	16544	32.2	15713	32.4	13903	32.4
foreman	22424	34.9	21289	35.2	14279	35.2
grandma	14808	34.7	14953	35.0	12308	35.0
hall	16504	33.2	15460	33.5	13934	33.5
missa	11024	38.7	11912	38.9	9700	38.9
mother	15656	34.6	14700	35.1	11991	35.1

Tab. 1: Number of bits and PSNR for intra coding

dependently, their parameters are hardly correlated and thus cannot be encoded efficiently. To increase the correlation, the node generation must take the parameters of the preceding frame into consideration. MBIC performs the node generation for inter frames in two steps. At first the motion estimator (Fig. 2) determines a coarse vector for each node of the previous frame, which is stored in the sampling point memory, using a block matching search. With these vectors, the sampling point generator creates an initial mesh. Starting from this mesh, the new node positions are determined by a node movement procedure as described in 2.2, and the node colours are optimized with respect to the overall MSE. For each node, the differences between the actual node parameters (position and colour) and the associated node parameters of the previous frame are calculated, and, after colour quantization, entropy encoded. The quantized differences are now added to the previous node parameters and stored in the sampling point memory as a prediction for the subsequent frame.

In addition to the change of the position or colour of nodes from frame to frame, MBIC is also able to insert or remove nodes. Thus, image contents can be handled that are not predictable by the preceding image.

4 Simulation Results

Simulations have been carried out to compare the performance of MBIC with a DCT-based and a wavelet-based video codec. For both intra and inter coding, the H.263 codec from Telenor, version 2.0, has been used with all special modes turned on, namely unrestricted motion vector mode, syntax-based arithmetic coding, advanced prediction mode and PB-frames mode.

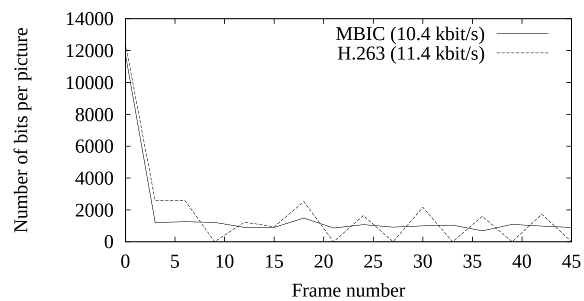

Fig. 8: Number of bits per picture for inter coding of *claire* sequence

Additionally for intra coding, the colour demo version of the wavelet codec SPIHT [7] has been included in the comparison. All considered video sequences have been processed in QCIF format (176 × 144 pels).

In **Tab. 1** the number of bits per picture and the peak signal to noise ratio (PSNR) for intra coding are given as a result. For each sequence the first picture has been coded with all coding schemes (H.263, SPIHT and MBIC). Their coding parameters have been adjusted to get approximately the same objective image quality (PSNR). As can be seen from the numbers of bits per picture, MBIC requires by far the lowest number of bits per picture.

In order to show the inter coding performance of MBIC, the *claire* sequence has been coded with the H.263 coder and the MBIC coder at 10 frames per second, i. e., each third frame of the original sequence has been processed. Both MBIC and the H.263 coder have been run without rate control. **Fig. 8** illustrates the number of bits per picture for both schemes.

As the H.263 coder operates with PB-frames, two successive pictures are coded jointly, i. e. for

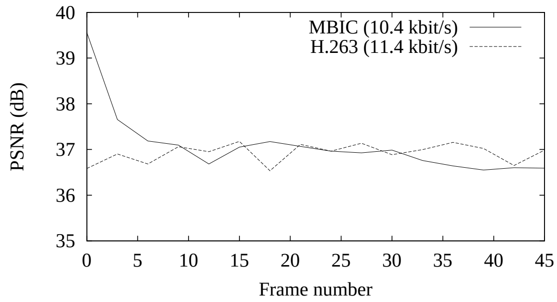

Fig. 9: Peak signal to noise ratio (PSNR) for inter coding of *claire* sequence

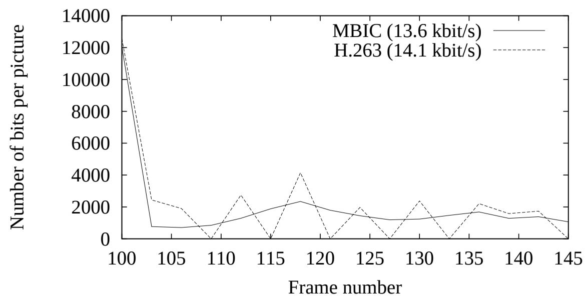

Fig. 10: Number of bits per picture for inter coding of *grandma* sequence

H.263 the numbers of bits in PB-frames 6, 18, 24, 30, 36 and 42 in Fig. 8 include two frames. Consequently the numbers of bits for frames 9, 21, 27, 33, 39 and 45 are zero. Leaving out the first intra frame, the calculated mean bit rate for MBIC is 10.4 kbit/s and 11.4 kbit/s for H.263. **Fig. 9** shows that MBIC obtains approximately the same PSNR at 10 % lower bit rate as the H.263 coder.

Fig. 10 shows the number of bits per picture for frame 100 to 145 of *grandma* sequence. The bit rate is 13.6 kbit/s for MBIC and 14.1 kbit/s for H.263. **Fig. 11** illustrates the PSNR for both schemes. For this sequence, the performance of both schemes is almost equal.

5 Conclusion

A new coding scheme MBIC (Mesh Based Interpolative Coding) using adaptive non-equidistant spatial sampling and interpolation has been proposed. It has been applied for coding of still images (intra coding) as well as for moving pictures (inter coding) at very low bitrates below 64 kbit/s. Simulation results have shown that MBIC can provide comparable picture quality (PSNR) to conventional schemes and requires less bit rate for sequences with moderate movement. There is room for improvements by using more complex sampling point generation algorithms for inter frame coding with MBIC.

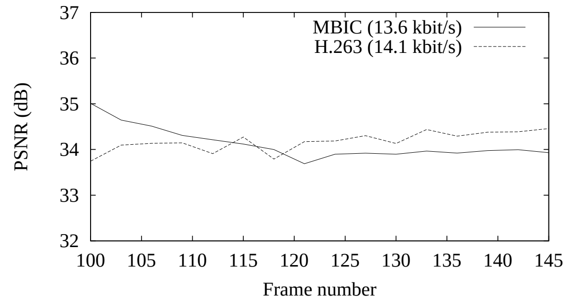

Fig. 11: Peak signal to noise ratio (PSNR) for inter coding of *grandma* sequence

References

[1] A. Netravali, B. Haskell: Digital Pictures – AT&T Bell Laboratories, 1988

[2] F. Preparata, M. Shamos: Computational geometry: an introduction – New York: Springer, 1993

[3] Y. Altunbasak, A. M. Tekalp: Occlusion-Adaptive, Content-Based Mesh Design and Forward Tracking – IEEE Trans. on Image Processing, Vol. 6, No. 9, Sept. 1997

[4] K. Schröder: Objektbeschreibende Bewegtbildcodierung mit gittergestützter Prädiktion – Dissertation Universität Dortmund, Fortschr.-Ber. VDI Reihe 10 Nr. 485, Düsseldorf, VDI-Verlag 1997

[5] E. Baum: Internal project report, 1999

[6] D. Salomon: Data compression: the complete reference – New York: Springer, 1998

[7] A. Said, W. Pearlman: A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees – IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 243-250, June 1996