# Novel video coding scheme using adaptive mesh based interpolation and node tracking

Eckhart Baum and Joachim Speidel

Institute of Telecommunications, University of Stuttgart
Pfaffenwaldring 47, 70569 Stuttgart, Germany

## ABSTRACT

An alternative method to H.263 for encoding of moving images at bit rates below 64 kbit/s is presented using adaptive spatial subsampling, mesh based interpolation and node tracking. This is why we call this new coding algorithm Mesh Based Interpolative Coding (MBIC). Data compression is achieved by representing the image content by a number of non-equidistant sampling points (nodes) for luminance and colour difference signals. The decoder reconstructs the image by interpolating the transmitted sampling points. For a given number of nodes which corresponds approximately to the total bit rate, the coder generates the node positions by minimizing the mean square error between the original picture and the interpolated picture. For moving images, each node is associated with a motion vector for node tracking. 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.

**Keywords:** image coding, spatial subsampling, adaptive interpolation, Delaunay mesh, node tracking

## 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. Each coded sample can have an arbitrary position and its position is automatically determined by a procedure, which minimizes the mean square error (MSE) between the original and the interpolated picture. In contrast to other interpolative techniques[1] the proposed algorithm MBIC uses a two-dimensional linear interpolation to reconstruct the area between the sampling points. For this purpose, all sampling points must be triangulated, i. e., a triangular mesh is constructed, whereby each sampling point forms one node of the mesh. Thus, the whole image area is covered by a number of non-overlapping triangles. The interpolated values of the pels within a triangle are defined by a linear combination of the three vertices belonging to the triangle. In order to get a unique triangular mesh in the coder and decoder without transmitting information about the topology of the mesh, a special class of triangular meshes is used, the so-called *Delaunay* meshes.

Inter frames are handled by moving the nodes from frame to frame and coding the corresponding vectors. Thus, only the changes between two frames must be coded. There is no need to transmit a difference picture like in other schemes, because it is possible to modify position and colour of each node arbitrarily. The decoder must store all node parameters of the preceding frame and applies the transmitted changes to reconstruct the current frame.

The principal functionality of the MBIC coder is shown in Fig. 1. At first, the sampling point generator SG generates a set $\mathbf{S}_j = \{\mathbf{s}_{1,j}, \mathbf{s}_{2,j}, \ldots, \mathbf{s}_{n_s,j}\}$ of $n_s$ sampling points for the incoming frame $j$ by an iterative procedure, as described in Section 2.2. For intra coding, all sampling points are initially set to a regular grid. For inter coding, an initial setting is found by using the sampling points $\hat{\mathbf{S}}_{j-1}$ of the preceding image, whereby each point is displaced according to a vector, which is determined by

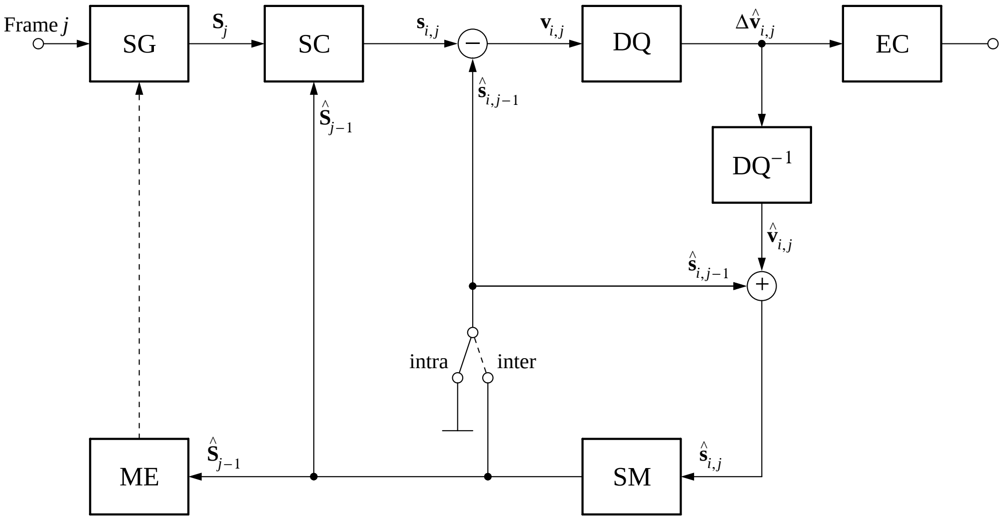



**Figure 1.** Principle block diagram of the MBIC coder — SG: sampling point generator, SC: scan, DQ: differential quantizer, EC: entropy coder, ME: motion estimator, SM: sampling point memory

a block motion estimation ME. The details are explained in Section 2.5. Starting from the initial setting, The positions $\mathbf{p}_{i,j}$ and colours $\mathbf{c}_{i,j}$ of each node $\mathbf{s}_{i,j} = (\mathbf{p}_{i,j}, \mathbf{c}_{i,j})$ are varied many times during the iteration procedure, in order to increase the overall quality of the interpolated picture. When the point is reached, where no further improvement can be achieved by varying the node parameters, the algorithm ends.

After that, the two-dimensionally placed sampling points are scanned along a certain path to get a one-dimensional list of consecutive sampling points $\mathbf{s}_{i,j}$, with $i \in \{1, 2, \ldots, n_s\}$. Details can be found in Section 2.3. The scanning is only performed in intra frames, while for all subsequent inter frames, the order of the sampling points remains the same. The parameters of the sampling points $\mathbf{s}_{i,j}$ are now differentially coded by subtracting the parameters of the corresponding quantized sampling points $\hat{\mathbf{s}}_{i,j-1}$ of the preceding frame $j-1$ for all $i \in \{1, 2, \ldots, n_s\}$. In other words, vectors $\mathbf{v}_{i,j} = \mathbf{s}_{i,j} - \hat{\mathbf{s}}_{i,j-1}$ are calculated, that describe the modifications of the sampling points from the previous to the current frame. For intra frames, $\mathbf{v}_{i,j}$ equals $\mathbf{s}_{i,j}$, as there is no predicting sampling point at that time.

The next block DQ subtracts the parameters of two successive vectors (resp. sampling points for intra coding) $\mathbf{v}_{i-1,j}$ and $\mathbf{v}_{i,j}$ and quantizes the difference, as described in Section 2.4. Finally, the quantized vector differences $\Delta\hat{\mathbf{v}}_{i,j} = (\mathbf{v}_{i,j} - \mathbf{v}_{i-1,j})_{\text{quant}}$, are entropy coded in EC using an arithmetic coder. In order to get the predicting sampling points $\hat{\mathbf{s}}_{i,j}$ for the following frame, the vector differences $\Delta\hat{\mathbf{v}}_{i,j}$ are summed in $DQ^{-1}$, and the resulting vectors $\hat{\mathbf{v}}_{i,j}$ are added to the sampling points $\hat{\mathbf{s}}_{i,j-1}$ of frame $j-1$. The new prediction is then stored in the sampling point memory SM, which has the capacity to hold all sampling points for one frame.

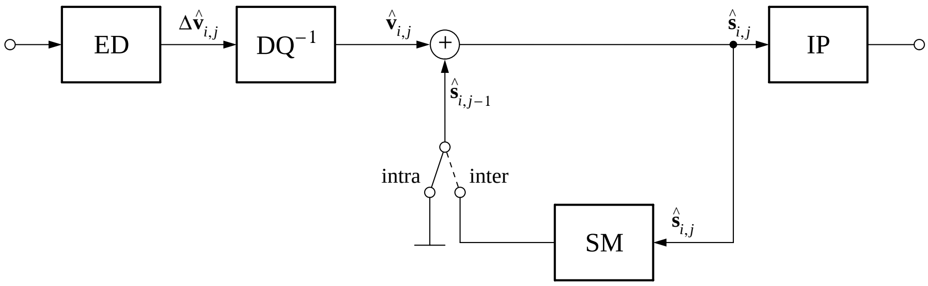



**Figure 2.** Principle block diagram of the MBIC decoder — ED: entropy decoder, DQ: differential quantizer, IP: interpolator SM: sampling point memory

Fig. 2 illustrates the block diagram of the MBIC decoder. First, the arriving bitstream is entropy decoded, which leads to the quantized vector differences $\Delta\hat{\mathbf{v}}_{i,j}$. Summing them up results in the quantized vectors $\hat{\mathbf{v}}_{i,j}$, which are added to the predicting sampling points $\hat{\mathbf{s}}_{i,j-1}$, to calculate the quantized sampling points $\hat{\mathbf{s}}_{i,j}$. The coded image is then reconstructed by interpolating the sampling points in IP, which is described in Section 2.1. Furthermore, the points are fed back to the sampling point memory SM, in order to form the prediction for the subsequent frame.

The paper is organized as follows: Section 2 describes the components of MBIC in detail, especially the mesh based interpolation, the node generation, the two-dimensional scanning of the nodes, the colour quantization, the entropy coding and the node tracking algorithm. Simulation results for intra and inter coding and a comparison with other image coding techniques are presented in Section 3. Section 4 gives a conclusion.

## 2. SPATIAL INTERPOLATIVE CODING

### 2.1. Interpolation

The interpolation forms the main component of the proposed coding system. Its task is to determine the values of the missing pels between the non-equidistant sampling points. This problem is also known as "scattered data interpolation" from approximation theory[2] and can be formulated as follows: Given a function $\mathbf{b}(x_i, y_i)$ and $n_s$ sampling points $\mathbf{s}_i = (\mathbf{p}_i, \mathbf{c}_i)$, $i \in \{1, 2, \ldots, n_s\}$, with arbitrary positions $\mathbf{p}_i = (x_i, y_i)$ and colours $\mathbf{c}_i = (Y_i, C_{R_i}, C_{B_i})$. A function $\tilde{\mathbf{b}}(x, y) = \left(\tilde{Y}(x, y), \tilde{C}_R(x, y), \tilde{C}_B(x, y)\right)$ shall interpolate the sampling points, so that $\tilde{\mathbf{b}}(x_i, y_i) = \mathbf{c}_i$ holds and $\tilde{\mathbf{b}}(x_i, y_i)$ approximates $\mathbf{b}(x_i, y_i)$. Because of its application to image coding, the interpolation function should have some special properties:

- The artifacts due to the interpolation should not be annoying for the human visual system. Therefore smooth colour transitions are preferred.
- One sampling point should not have influence on all image pels, but only on a locally restricted area of the image. The restricted influence is suited to the characteristics of natural images, which consist of several strictly bounded objects. It would not make any sense, that a sampling point within an object affects the colour outside of this object. Furthermore, when changing a sampling point, only the affected part of the image must be interpolated, which saves much computing time.
- The costs of the interpolation should be as small as possible, as the sampling point generator frequently performs interpolations, because of its iterative technique, as described in Section 2.2.

Due to these requirements, a two-dimensional linear interpolation method, based on triangular meshes, has been chosen. The resulting interpolation function is piecewise linear and everywhere steady. To derive the necessary equations, Fig. 3 illustrates the interpolated luminance and colour difference signals $\tilde{Y}(x, y)$, $\tilde{C}_R(x, y)$ and $\tilde{C}_B(x, y)$ for three given sampling points $\mathbf{s}_1 = (\mathbf{p}_1, \mathbf{c}_1)$, $\mathbf{s}_2 = (\mathbf{p}_2, \mathbf{c}_2)$ and $\mathbf{s}_3 = (\mathbf{p}_3, \mathbf{c}_3)$.

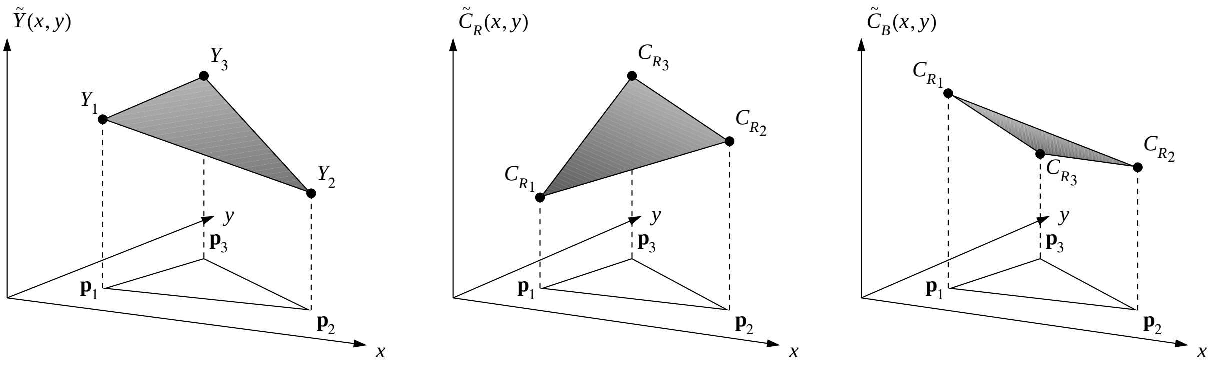



**Figure 3.** Two-dimensional linear interpolation of the luminance and colour difference signals $Y$, $C_R$ and $C_B$

Geometrically viewed, the sampling points span three planes in the three-dimensional space. Each colour within the triangular area can be calculated by the following linear combination:

$$\tilde{\mathbf{b}}(x, y) = \lambda_1(x, y)\,\mathbf{c}_1 + \lambda_2(x, y)\,\mathbf{c}_2 + \lambda_3(x, y)\,\mathbf{c}_3 \tag{2.1}$$

with

$$\lambda_1(x, y) = \frac{y_2 - y_3}{\Delta} x + \frac{x_3 - x_2}{\Delta} y + \frac{x_2 y_3 - x_3 y_2}{\Delta} \tag{2.2}$$

$$\lambda_1(x, y) = \frac{y_3 - y_1}{\Delta} x + \frac{x_1 - x_3}{\Delta} y + \frac{x_3 y_1 - x_1 y_3}{\Delta} \tag{2.3}$$

$$\lambda_1(x, y) = \frac{y_1 - y_2}{\Delta} x + \frac{x_2 - x_1}{\Delta} y + \frac{x_1 y_2 - x_2 y_1}{\Delta} \tag{2.4}$$

and

$$\Delta = x_1 y_2 + x_2 y_3 + x_3 y_1 - x_1 y_3 - x_2 y_1 - x_3 y_2 \tag{2.5}$$

If more than three sampling points are given, they have to be triangulated, in order to obtain a triangular mesh, whereby each sampling point forms one node of the mesh. Thus the whole image area is covered by non-overlapping triangles. The mesh is fully connected, i. e., there is no pair of nodes, which could be connected additionally without crossing an existing edge. Fig. 4 shows an example with seven sampling points (1) and a corresponding triangular mesh (2).

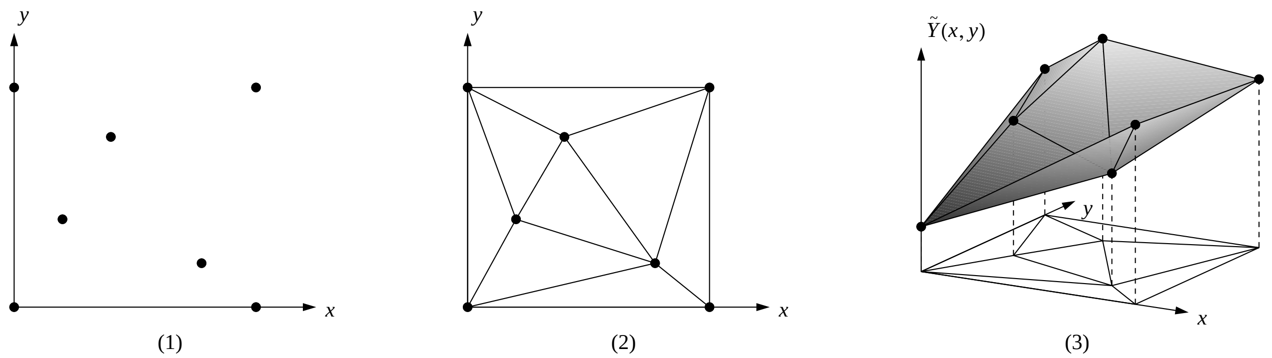



**Figure 4.** Triangulation of a set of sampling points

Each pel within the image, that does not lie on an edge of the mesh, belongs to exactly one triangle. Consequently, its value can be interpolated by the equations (2.1) - (2.5). A pel, that lies exactly on an edge connecting two nodes, belongs to both triangles, which have the edge in common. In this case, assigning the point to a triangle is ambiguous. However, it does not matter which triangle to use for interpolation, because the interpolated value only depends on the two nodes, but not on the third node that would be different for both triangles. Fig. 4 (3) illustrates the interpolated luminance signal $\tilde{Y}(x, y)$ for the sampling points of Fig. 4 with arbitrarily chosen grey levels.

As the interpolation procedure is not only implemented in the decoder, but also in the node generator of the coder, both coder and decoder should produce the same interpolated image for a given set of nodes. This requires the same interplation rules Eq. (2.1) - (2.5) 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 $\mathbf{s}_1$ and $\mathbf{s}_3$ are assumed to be black, while the pels at nodes $\mathbf{s}_2$, $\mathbf{s}_4$ and $\mathbf{s}_5$ are assumed to be white. Obviously, there are several possibilities to triangulate a given set of nodes. This implicates differences between the resulting pictures.

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 implemented node generation algorithm, 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*[3], which is therefore used by MBIC. Delaunay triangulations have also been used for motion compensation as an alternative to block matching[4,5].

One property of the Delaunay triangulation, which is used for constructing the mesh with an incremental insertion algorithm[6], concerns the outcircles of the triangles: For every triangle $T(\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k)$ belonging to a Delaunay triangulation of a set of nodes $\mathbf{S}$, the circle defined by $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$ contains no other node of $\mathbf{S}$. Fig. 6 shows two triangulations for the same node positions. The

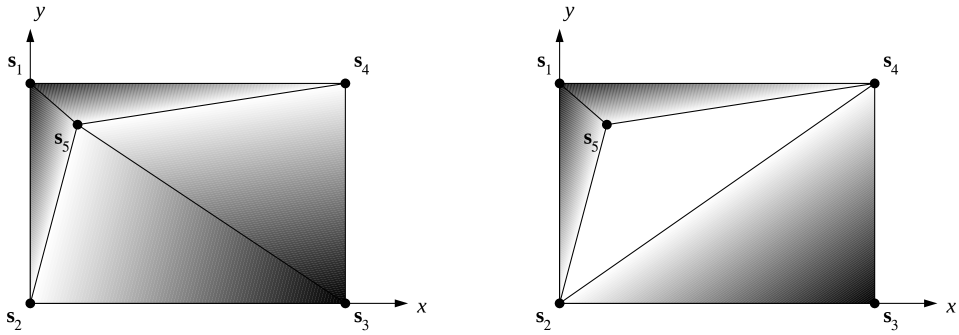



**Figure 5.** Two possible triangulations for a given set of nodes

left example fulfills the Delaunay criterion, while the right does not, because the circle defined by $\mathbf{s}_1$, $\mathbf{s}_3$ and $\mathbf{s}_4$ contains $\mathbf{s}_5$.

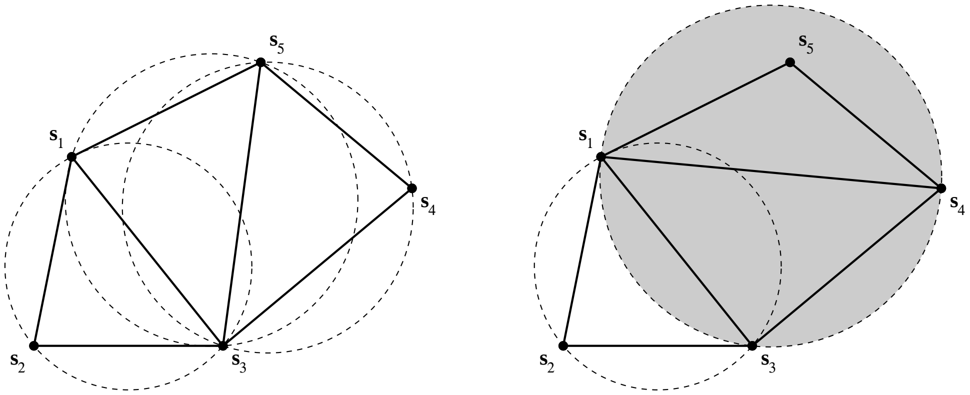



**Figure 6.** Two triangulations and the outcircles of the triangles

## 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 suitable positions for all nodes, MBIC performs an iterative procedure based on node movement, deletion and insertion. Fig. 7 shows the principle block diagram of the implemented node generation.

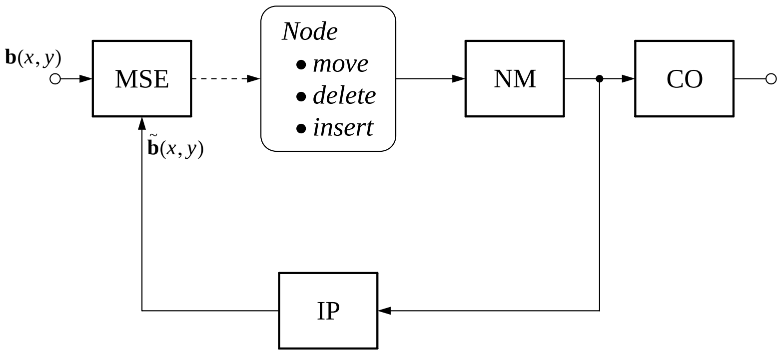
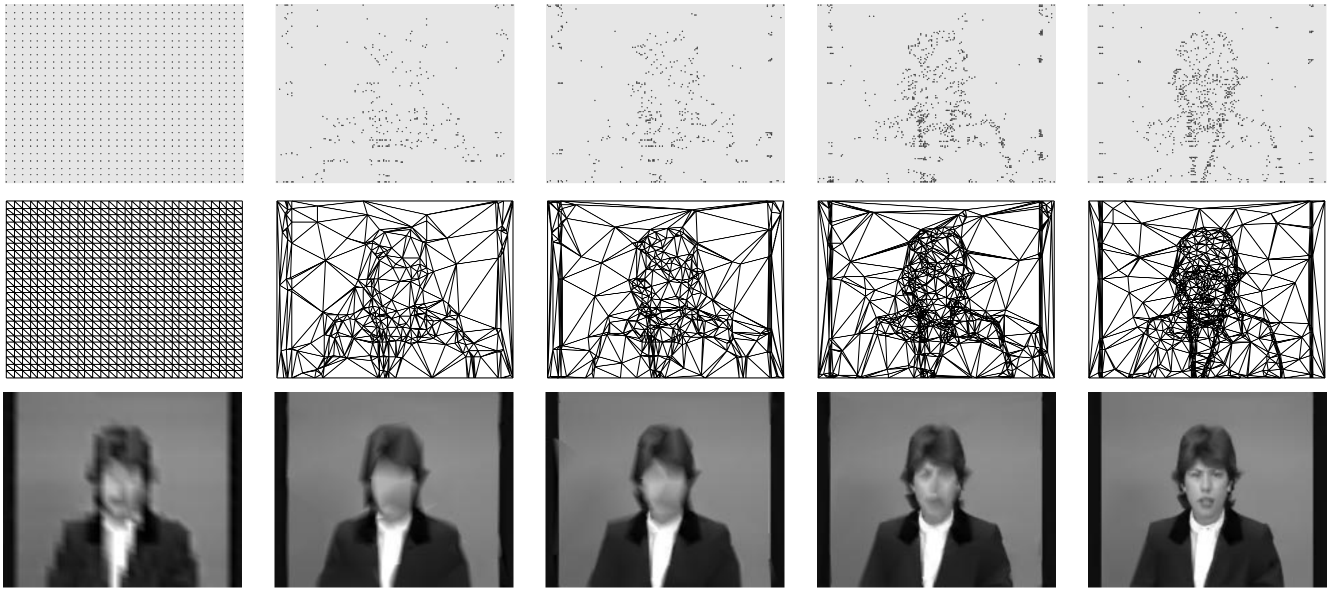



**Figure 7.** Principle block diagram of the node generation — MSE: mean square error evaluation, NM: node memory, CO: colour optimizer, IP: interpolator

At the beginning, the node memory NM contains a set of regular distributed nodes, as shown in Fig. 8 (1). Starting from these nodes and the appropriate mesh, 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 interpolated picture $\tilde{\mathbf{b}}(x, y)$ and the original picture $\mathbf{b}(x, y)$. $a = 1$ has shown to be effective[7]. Nodes are removed, if their removal does not result in a MSE of the changed triangles which exceeds an upper limit, and additional nodes are inserted into triangles, so that the MSE is reduced furthermore. The allowed node positions are restricted to the pel raster. Fig. 8 (2) illustrates the positions of the nodes in the node memory NM, the mesh and the interpolated image after the first iteration step. The same steps are carried out sequentially as shown in Fig. 8 (3) – (5). The implemented algorithm always converges to a state, where no more changes are to be applied.



(1) initial setting (2) 1st iteration (3) 2nd iteration (4) 4th iteration (5) 9th iteration

**Figure 8.** Iteration steps

During the iteration, the colour of 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.

## 2.3. Node Scanning

After generating the sampling points, each point has an arbitrary position in the two-dimensional space. For transmitting them successively to the receiver, at first they must be arranged in a sequential order. Neighbouring nodes have similar positions and often similar colours. To benefit from the correlation between these nodes, the transmission order should retain the neighbourhood of the nodes. In this case, a differential encoding of the nodes can be performed effectively. There are many possibilities for the sequential ordering. One method, that has shown to be very efficient, scans all pel positions along a *Hilbert curve*[8], which requires that all nodes lie on integer pel positions.

To demonstrate the characteristic of a Hilbert curve, its recursive construction principle for a grid of $16 \times 16$ pels is shown in Fig. 9. As can be seen, a coarse path through a $2 \times 2$ grid (1) is refined for a grid with the double resolution (2). The new path is refined furthermore (3) and (4), until the desired resolution is reached. This construction principle makes clear, that nodes, which lie closely together in the two-dimensional space, are also close together in scan order. Therefore the positions and colours of successive nodes are highly correlated. To create a Hilbert path for the QCIF image size ($176 \times 144$ pels), $11 \times 9$ Hilbert curves with each $16 \times 16$*pels* are merged.

The scan is only performed for intra coded pictures. For inter coded pictures, the nodes are generated starting from the nodes of the preceding picture (Section 2.5). Hence there are fixed correspondences between the new (displaced) nodes and the preceding nodes, which makes it possible to encode the node changes $\mathbf{v}_{i,j} = \mathbf{s}_{i,j} - \hat{\mathbf{s}}_{i,j-1}$ from frame $j - 1$ to frame $j$. Maintaining these correspondences means not to change the order of the nodes, i. e., the nodes must not be scanned in inter coded pictures.

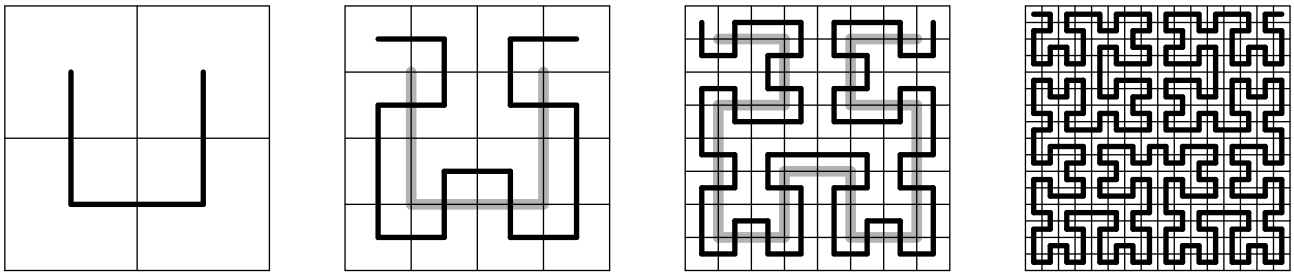



**Figure 9.** Recursive construction principle of the Hilbert curve

### 2.4. Quantization and Entropy Coding

To take advantage of the correlated nodes for intra coding, the positions and colours of successive nodes are differentially encoded by predicting each sampling point $\mathbf{s}_{i,j} = (\mathbf{p}_{i,j}, \mathbf{c}_{i,j})$ by its predecessor $\mathbf{s}_{i-1,j}$ in scan order. Accordingly, for inter coding each vector $\mathbf{v}_{i,j} = (\Delta\mathbf{p}_{i,j}, \Delta\mathbf{c}_{i,j})$ is predicted by its predecessor $\mathbf{v}_{i-1,j}$ to get the difference vector $\Delta\mathbf{v}_{i,j} = (\Delta\mathbf{p}_{i,j} - \Delta\mathbf{p}_{i-1,j}, \Delta\mathbf{c}_{i,j} - \Delta\mathbf{c}_{i-1,j}) = (\Delta^2\mathbf{p}_{i,j}, \Delta^2\mathbf{c}_{i,j})$, as shown in Fig. 1 ($\mathbf{v}_{i,j} = \mathbf{s}_{i,j}$ and $\Delta\mathbf{v}_{i,j} = \mathbf{s}_{i,j} - \mathbf{s}_{i-1,j}$ holds for intra coding). To increase the coding efficiency, the colour differences $\Delta^2\mathbf{c}_{i,j}$ are quantized using an adaptive quantization. It has been shown[9], that the colour quantization of nodes lying closely together is less critical than for distant nodes, because the image area, which is affected by the quantization of a node, decreases when the node density becomes higher. Hence, each colour difference $\Delta^2\mathbf{c}_{i,j}$ is quantized by an individual factor, that depends on the area of the triangles surrounding the node $\mathbf{s}_{i,j}$. The quantizer scales, that have been used by the coder, can also be determined by the decoder itself, so they have not to be transmitted. The position differences $\Delta^2\mathbf{p}_{i,j}$ may not be quantized, because even small modifications of a node position could cause a change of the mesh topology, which likely results in a severe image degradation.

After calculating the quantized difference vectors $\Delta\hat{\mathbf{v}}_{i,j} = (\Delta^2\mathbf{p}_{i,j}, \Delta^2\hat{\mathbf{c}}_{i,j})$, they are arithmetically coded[10], with different frequency tables for position, luminance and chrominance. For intra coding, the position of a node is coded by its distance (in pel steps) from the preceding node along the Hilbert path, as this has shown to be more efficient than coding *x* and *y* coordinates.

### 2.5. Node Tracking

So far, all described methods concerned mainly the coding of a single image. In order to benefit from the correlation of successive frames in image sequences, the frames are inter coded, i. e., only the changes from frame to frame are coded. As described, in MBIC a picture is represented by a number of nodes with parameters position and colour. For moving images, both 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 independently, 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 a motion estimation (ME in Fig. 1) determines a coarse motion vector for each node $\hat{\mathbf{s}}_{i,j-1}$ of the previous frame, using a block matching algorithm. A $15 \times 15$ pel sized block of the previous frame, whose centre has the position of the node, is searched in the current frame by minimizing the square error over all positions within the search range.

Applying these motion vectors to the node positions of the previous frame, the sampling point generator creates an initial mesh. Starting from this mesh, the final node positions $\mathbf{p}_{i,j}$ are determined by an iterative node movement procedure as described in Section 2.2. The nodes are repeatedly moved up to $\pm 1$ pixel in any direction, one at a time, to find an optimum position with regard to rate and distortion[11]. At last, the node colours $\mathbf{c}_{i,j}$ are optimized with respect to the overall MSE. This leads to the new nodes $\mathbf{s}_{i,j}$, which differ from their appropriate predecessors by $\mathbf{v}_{i,j} = \hat{\mathbf{s}}_{i,j} - \hat{\mathbf{s}}_{i,j-1}$.

In addition to the change of node positions or colours 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. The decision to insert or remove a node is reached in the same way as explained in Section 2.2.

## 3. 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. Additionally for intra coding, the colour demo version of the wavelet codec SPIHT[12] 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.

**Table 1.** Number of bits and PSNR for intra coding

| 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 |

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. 10 illustrates the number of bits per picture for both schemes.

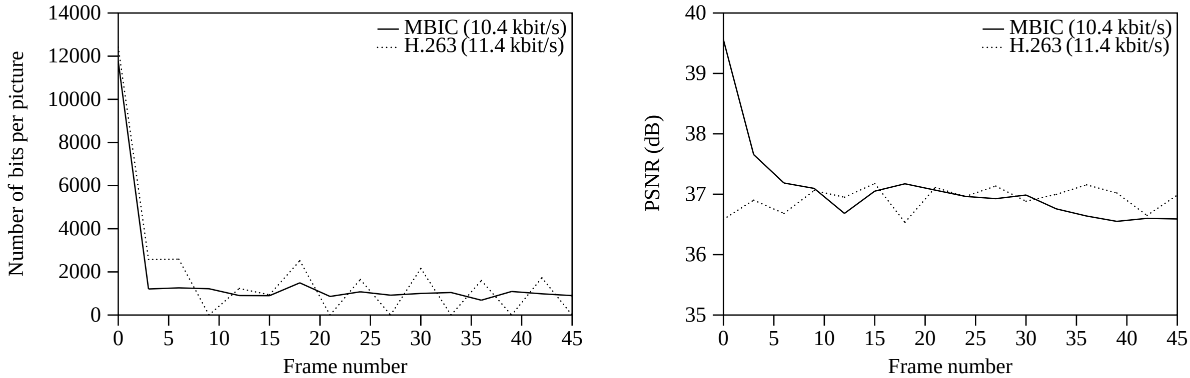



**Figure 10.** Number of bits per picture and peak signal to noise ratio (PSNR) for inter coding of *claire* sequence

As the H.263 coder operates with PB-frames, two successive pictures are coded jointly, i. e. for H.263 the numbers of bits in PB-frames 6, 18, 24, 30, 36 and 42 in Fig. 10 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. 10 shows that MBIC obtains approximately the same PSNR at 10 % lower bit rate as the H.263 coder.

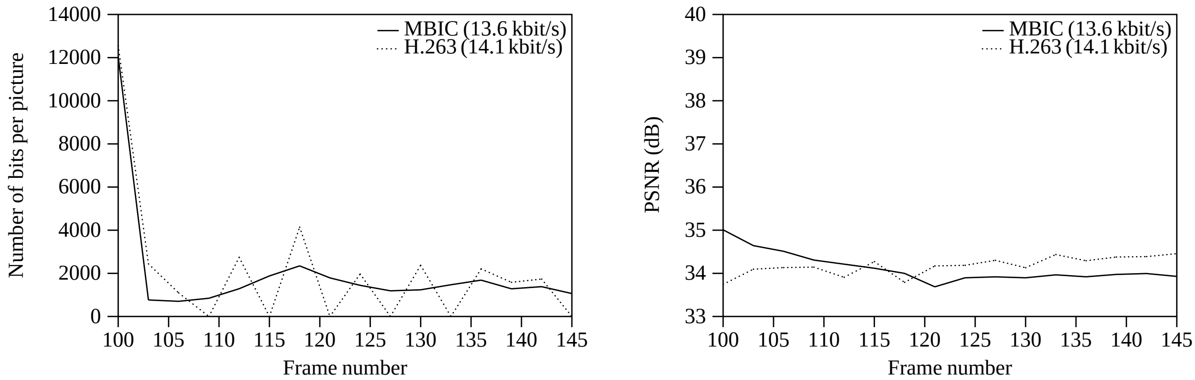



**Figure 11.** Number of bits per picture and peak signal to noise ratio (PSNR) for inter coding of *grandma* sequence

Fig. 11 (left) 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 (right) illustrates the PSNR for both schemes. For this sequence, the performance of both schemes is almost equal.

## 4. 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 for intra coding have shown, that the spatial interpolation technique is well suited to coding of natural images, as it provides a good subjective and objective image quality at high compression rates. For inter coding, 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.

## REFERENCES


1. A. Netravali, B. Haskell. *Digital Pictures.* AT&T Bell Laboratories, 1988.
2. L. Schumaker. "Fitting Surfaces to Scattered Data". In *Approximation Theory II*, pages 203–268. Academic Press, 1977.
3. F. Preparata, M. Shamos. *Computational geometry: an introduction.* New York: Springer, 1993.
4. Y. Altunbasak, A. M. Tekalp. "Occlusion-Adaptive, Content-Based Mesh Design and Forward Tracking". *IEEE Trans. on Image Processing* **6**, 1270–1280 (Sept. 1997).
5. K. Schröder. *Bewegtbildcodierung mit gittergestützter Prädiktion.* Ph.D. thesis, Fortschr.-Ber. VDI Reihe 10 Nr. 485, Düsseldorf, VDI-Verlag, 1997.
6. C. Lawson. "Software for $C^1$ Surface Interpolation". In *Math. Software III*, pages 161–194. Academic Press, 1977.
7. E. Baum. *Internal project report*. Institute of Telecommunications, University of Stuttgart, 1999.
8. D. Salomon. *Data compression: the complete reference.* New York: Springer, 1998.
9. C. Kärner. *Quantisierung von Stützstellen bei der Bildcodierung mit adaptiver Interpolation.* Semester thesis, Institute of Telecommunications, University of Stuttgart, 1997.
10. I. Witten, R. Neal, J. Cleary. "Arithemtic Coding for Data Compression". *Comm. of the ACM* **30**, 520–540 (June 1987).
11. T. Berger. *Rate-Distortion Theory. A Mathematical Theory for Data Compression.* Prentice-Hall, 1971.
12. 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* **6**, 243–250 (June 1996).