VERY LOW BIT RATE VIDEO CODING USING ACTIVE TRIANGULAR MESH

Philippe Salembier *Xavier Ayuso*

Dept. of Signal Theory and Communications
ETSETB - Universitat Politècnica de Catalunya
Campus Nord - Módulo D5
C/ Gran Capità, 08034 Barcelona, Spain
E-mail: philippe@gps.tsc.upc.es

ABSTRACT

This paper presents a video coding scheme for very low bit rate applications. The coding approach relies on the notion of *active meshes* and can be viewed as a particular case of region-based coding. The active mesh is used to efficiently represent and code the various regions of the scene as well as the motion information. The algorithm involves the following steps: Projection (time tracking of the regions), Partition and Motion coding, Synthesis, Modification of the mesh topology (introduction of new regions and elimination of useless regions) and texture coding.

1. INTRODUCTION

In the framework of very low bit rate video coding, there is an increasing interest in second generation image compression techniques [2]. In particular, region-based compression methods describe the images in terms of a set of regions, that is a partition, and of some information for each region to be used by the receiver to reconstruct the image. For sequences, region-based schemes have been developed in particular in [3, 4, 7, 5].

The main difference between the techniques proposed in [3, 4, 7, 5] can be described by the relative importance they assign to the spatial or the motion information. Reference [3] proposes a coding scheme where motion plays the central role and the image is restored on the receiver side by motion compensation of past restored frames. A partition is used mainly to define the regions that should be compensated. This approach leads to good results if the sequence can actually be compensated. That is, if no new objects are allowed to be introduced in the scene and if scene changes are prohibited. As a result this technique is mainly dedicated to very specific applications such as "head & shoulders" sequences. The approaches described in [4, 7, 5] are more general in the sense that they mainly deal with the spatial information of the scene.

In [5], a coding algorithm combining a spatial analysis of the sequence with a motion compensation of the transmitted information was proposed. On the one hand side, the spatial analysis is used to get a general scheme able to deal with any kind of sequences and scene changes. On the other hand side, motion information is used to increase the coding efficiency by compensation of the spatial information that has to be transmitted (partition and texture). An analysis of the results given in [5] show that 55% of the bitstream is devoted to the partition information, 35% to the texture and the remaining 10% to motion.

The goal of this paper is to present an improved scheme which follows the same philosophy as the one proposed in [5] but where much more texture information is sent to the receiver. To reach this goal, we propose to work not with arbitrary shaped regions as in [5] but with simple shapes as triangles. Moreover, we would like to send at the same time the motion and the partition information. This idea leads to the definition of an active triangular mesh coding scheme.

Active meshes are intensively studied in the framework of coding [6, 1]. They are generally used to model the motion information but very few complete active mesh coding schemes have been proposed. In our proposal, the mesh is used to model the sequence itself and plays two fundamental roles: First, it defines a simplified partition made of triangles. The partition is used to define homogeneous regions in terms of texture. Because of its particular structure, the partition itself can be very efficiently coded. Second, the nodes of the triangular mesh are also used to send the motion information. This means that the motion of each region is assumed to be modeled by an affine transformation and can take into account rather complex events.

The organization of this paper is as follows: The following section describes the structure of the encoding algorithm. The main steps of the algorithm will be analyzed in details in section 3. Finally, section 4 presents some coding results.

2. ACTIVE TRIANGULAR MESH CODING SCHEME

The main structure of the algorithm can be seen in Fig.1. The structure has strong similarities with the one defined in [5]:

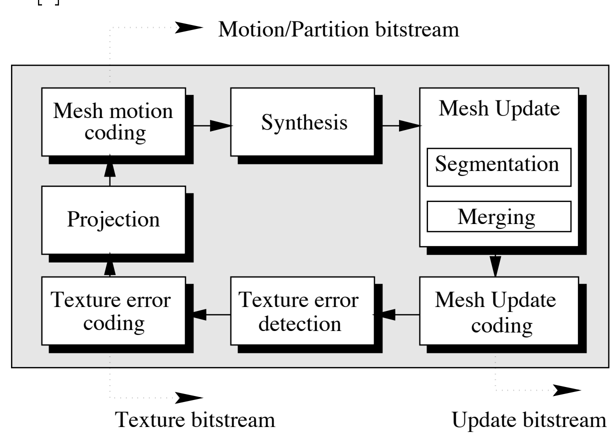


Figure 1. Structure of the Active Mesh coding scheme

- The "**projection**" step defines the time evolution of the mesh. It achieves at the same time the definition of the new partition (made of triangles) and the motion

*This work was supported by the European Community through the MORPHECO Project of the RACE program

estimation (motion of nodes). As in [5], this step does not allow the introduction of new regions.

- The "**Mesh motion coding**" block codes the information necessary to modify the mesh on the receiver side. This information consists in a displacement vector for each node of the mesh.
- The texture compensation is called "**Synthesis**". The time evolution of each triangle is defined by the motion of its three vertices. This defines an affine motion model which is used to compensate the texture.
- The goal of the "**Mesh update**" is two fold: First, new objects or strong texture inhomogeneities may have appeared. In order to restore the texture homogeneity, some triangles should be split. This can be viewed as a segmentation process which produces a denser triangular mesh. Second, some triangles may be useless because they have a very small area or because they are extremely elongated. Therefore, some nodes should be eliminated and a triangularization algorithm is applied to restore the triangular structure of the mesh.
- "**Mesh Update coding**": The information about the modification of the mesh topology defined by the previous block (nodes introduction or elimination) are coded and sent to the receiver.
- As mentioned before, the texture within some triangles maybe of poor quality. The objective of the "**Texture error detection**" block is to select the smallest number of triangles where the texture has to be improved.
- The quality of the selected triangles is improved by coding their texture. This step is achieved in the "**Texture error coding**".

In the sequel, the various steps of the algorithm are more precisely described.

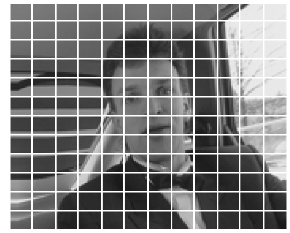
3. DESCRIPTION OF THE VARIOUS STEPS OF THE ALGORITHM

3.1. Mesh initialization

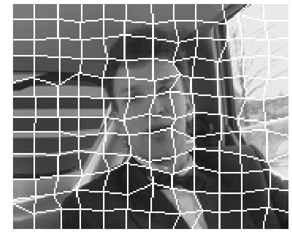
The first step of the algorithm is to define a mesh that is adapted to the image content. The definition of the mesh for the first frame is illustrated in Fig.2. The initial mesh is a regular square mesh as shown in Fig.2.a. The nodes of this mesh are displaced to be located on high gradient points. This is done by a simple search algorithm similar to the block matching algorithm working on the morphological gradient of the image (see Fig.2.b). Then, each resulting polygon is transformed into two triangles (see Fig.2.c). Since the polygons have four nodes, there are two possible ways of splitting them into two triangles. The splitting leading to the lowest variance within each resulting triangle is selected. Finally, nodes are removed in non-active areas of the image. This breaks the triangular structure of the mesh which has to be restored by a triangulation algorithm (for example a Delaunay triangulation). This procedure gives the initial mesh shown in Fig.2.d. As can be seen, the mesh is matched to the frame to code: the density of triangles depends on the signal activity and the edges of the mesh follow the strong gradient of the image.

3.2. Projection & Motion coding

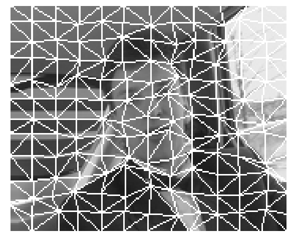
The objective of the projection is to follow the time evolution of the sequence. The mesh is modified by a displacement of its nodes but no triangle is allowed to appear or to disappear. The projection is done in two steps: the first step estimates the motion of each node and the second step defines the new mesh based on the motion of the set of nodes.



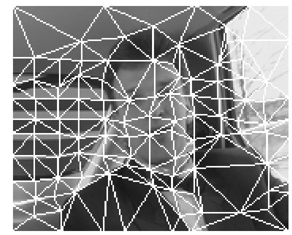
a) Regular square mesh



b) Modif. of the nodes position



c) Non-homog. square mesh



d) Non-homog. triangular mesh

Figure 2. Initialization of the mesh

- *Motion estimation*: In this first step, the motion of each node is estimated by a block matching. The search area can be rather large because the estimation only involves a small amount of nodes (typically 100). We have used a certainty criterion to validate the estimation: if the displaced frame difference (DFD) at the optimal position is not significantly lower than the mean of the DFD for all position of the search area, then the estimation is discarded and a zero motion is assumed.
- *Mesh definition*: Once the displacement vectors of the nodes have been computed, the new mesh should be defined. Note that the procedure is not straightforward because if all nodes are displaced accordingly to the estimated vectors, the resulting structure may not be a triangular mesh. For example, one has to prevent one vertex of a triangle to cross the opposite edge of the triangle. To solve this problem, the set of restrictions to be used to preserve the mesh topology has been studied and an iterative algorithm that achieves a constrained migration of the nodes has been defined.

The projection has defined a translation vector for each node of the mesh. This information has to be sent to the receiver in order to create the new mesh, that is the new partition. However, this information is also used to define the motion field to compensate the texture information. As can be seen, the "motion" and "partition" bitstreams that were present in the segmentation-based scheme of [5] are now merged into a single bitstream.

3.3. Synthesis

The synthesis or compensation of the image makes a prediction of the current frame based on the knowledge of the nodes motion. As mentioned in the previous section, the nodes motion allows a restoration of a dense motion field within each triangle. Indeed, the time evolution of each triangle is characterized by the displacement of its three nodes. This defines a geometrical affine transformations with six independent parameters.

The compensation is performed in a backward mode as illustrated by Fig.3. Each pixel (i, j) of the triangle of frame T is transformed in the pixel (x, y) of frame $T-1$. Since the coordinate of (x, y) are generally real values, a bilinear interpolation is used to define the gray level value that will be assigned to the pixel (i, j). The mode of compensation is backward in the sense that, the values of pixels of frame T are defined by computing the values at pixels location of frame $T - 1$. This backward mode offers the advantage of assigning one value to each pixel of frame T.

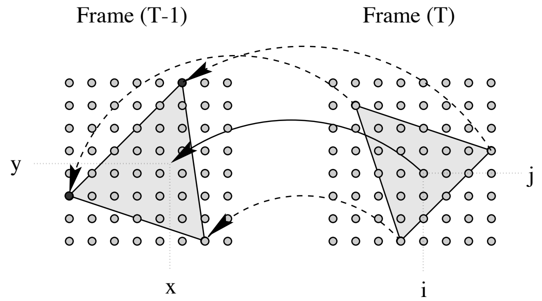


Figure 3. Synthesis of the texture

3.4. Modification of the mesh topology

In order to efficiently code the sequence over a large number of frames, the nodes of the mesh should follow the evolution of the scene, but the topology of the mesh should also be modified. Indeed, because of the modifications of the scene content, new triangles may be necessary and, by contrast, some triangles may degenerate and should be removed. This is the purpose of the "Segmentation" and "Merging" steps.

3.4.1. Segmentation

"Segmentation" means that we want to introduce new triangles in the mesh in order to more accurately represent the texture or the motion. To select the triangles that have to be segmented, two different criteria are used:

- Gradient criterion: if a high gradient component of the signal crosses one edge of a triangle, this triangle should be split in such a way that one new edge will follow the high gradient.
- Geometrical criterion: In practice, a triangle of very large size is very likely to produce large synthesis errors. It should be segmented into smaller triangles.

Three splitting strategies shown in Fig.4 have been defined. As can be seen, they either segment the original triangle in two or four. A careful analysis of the system behavior has shown that, in general, new nodes should not be introduced in the inside of the triangle because they will create a large number of useless triangles. In our scheme, the new nodes are always located on existing edges and on high gradient positions.

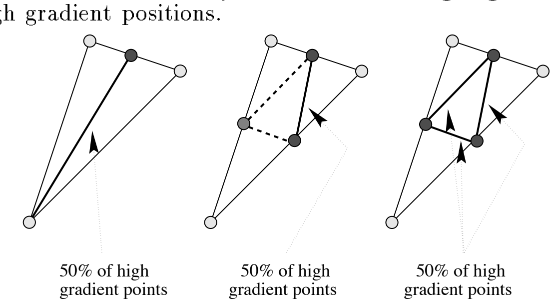


Figure 4. Segmentation of triangles

When a new node has been introduced to split one triangle, several new edges should be created to preserve the mesh structure. As shown in Fig.5, all polygons of the mesh should have only three vertices.

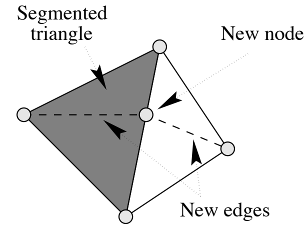


Figure 5. Creation of new edges

3.4.2. Merging

The process of node elimination is illustrated in Fig.6. It has to be used to remove triangles that are either very small or degenerated, that is, triangles with one edge of size much smaller than the two others. In the left side of Fig.6, two degenerated triangles can be seen. For each small or degenerated triangle, the node corresponding to the smallest gradient is removed. This node extraction destroys the mesh topology as can be seen in the center of Fig.6. Therefore, a triangulation algorithm has to be used to add new edges and restore the triangular topology of the mesh.

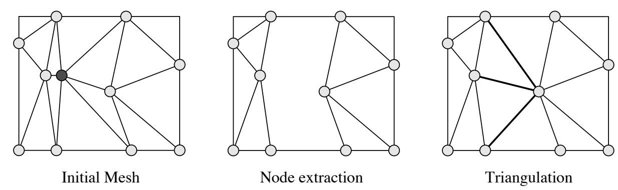


Figure 6. Elimination of triangles and Triangulation process

The information about the node introduction or extraction is sent to the receiver. This information can be efficiently coded: in the case of node introduction, we have seen that the new node is located on an existing edge. This restriction limits strongly the entropy of the information. Moreover, for the node elimination, one has simply to indicate which nodes are removed since the triangulation algorithm is purely geometrical and can be performed on the receiver side.

3.5. Texture coding

The texture coding involves, as a first step, the detection of areas where the synthesis has produced a poor representation of the image. Fig.7.a and 7.b respectively show the previous frame and the synthesized frame. The difference between the synthesized frame and the original current frame is computed and analyzed to extract a mask indicating where the texture has to be improved (see Fig.7.c: zones of high error and 7.d: Error mask). Fig.7.e shows the coded frame.

For the texture coding itself, classical region-based texture coding techniques as the ones reported in [5] are used.

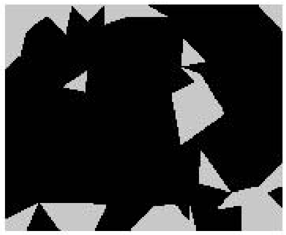
4. RESULTS

Table 1 shows some examples of coding results. The figures are averages obtained on 200 frames and the bitrates correspond to 5 Hz of frame rate. This active mesh scheme actually allows the coding of the shape and motion information at a reduced cost and the analysis of the bitstreams reveal that more than 80% of the bits are devoted to texture coding. Moreover, the sequence quality is better than the results reported in [5] for the same bitrates. Examples of coded frames are shown in Fig.8, and 9.

a) Frame at time $T-1$

b) Synthesized frame

c) Zone of high error



d) Polygonal error mask

e) Coded frame

Figure 7. Texture coding process

Sequence	Bits per frame	Rate kbits/s	PSNR dB
Miss America	2067	10.7	34.5
Carphone	5561	27.8	29.5
Mother & Daughter	3163	15.8	29.7
Foreman	7253	36.2	29.5

Table 1. Examples of coding results

Figure 8. Example of coding results on *Carphone*

Figure 9. Example of coding results on *Mother & Daughter*

5. CONCLUSIONS

In this paper, we have shown how to define an active mesh coding scheme that follows the strategy of more classical region-based scheme. The mesh is considered in a first step as an efficient way to code the partition of the image. Moreover, the mesh handles both the partition and the motion information. This allows a significant saving of bits that can be used texture coding.

The mesh is considered as the basic partition that represents the sequence. It is therefore necessary to be able to follow the time evolution of the mesh (the *projection* step) and to allow possible modifications of the mesh topology (the *update* step). Note that this coding strategy actually tracks regions in time and opens the door to content-based manipulation of images.

The texture is coded by a prediction based on an affine compensation (synthesis) of the previously coded image and the compensation error is coded by region-based coding techniques.

This coding approach gives very promising results compared to classical region-based scheme because it very efficiently deals with the motion and partition information.

REFERENCES

[1] M. Dudon, O. Avaro, and C. Roux. Triangle-based motion estimation and temporal interpolation. In IEEE, editor, *1995 IEEE workshop on Nonlinear Signal and Image Processing*, pages 242–245, Halkidiki, Greece, June 20-22 1995.

[2] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second generation image coding techniques. *Proceedings of the IEEE*, 73(4):549–575, April 1985.

[3] H.G. Musmann, M. Hőtter, and J. Ostermann. Object-oriented analysis-synthesis coding of moving images. *Signal Processing, Image Communication*, 1(2):117–138, October 1989.

[4] S. Rajala, M. Civanlar, and W. Lee. Video data compression using three-dimensional segmentation based on HVS properties. In IEEE, editor, *International Conference on Acoustics, Speech and Signal Processing*, pages 1092–1095, New York (NY), USA, 1988.

[5] P. Salembier, L. Torres, F. Meyer, and C. Gu. Region-based video coding using mathematical morphology. *Proceedings of IEEE (Invited paper)*, 83(6):843–857, June 1995.

[6] Y. Wang and O. Lee. Active mesh - a feature seeking and tracking image sequence representation scheme. *IEEE Transactions on Image Processing*, 3(5):610–624, September 1994.

[7] P. Willemin, T. Reed, and M. Kunt. Image sequence coding by split and merge. *IEEE Transactions on Communications*, 39(12):1845–1855, December 1991.

VERY LOW BIT RATE VIDEO CODING USING ACTIVE TRIANGULAR MESH

Philippe Salembier and Xavier Ayuso

Dept. of Signal Theory and Communications
ETSETB - Universitat Politècnica de Catalunya
Campus Nord - Módulo D5
C/ Gran Capità, 08034 Barcelona, Spain
E-mail: philippe@gps.tsc.upc.es

This paper presents a video coding scheme for very low bit rate applications. The coding approach relies on the notion of *active meshes* and can be viewed as a particular case of region-based coding. The active mesh is used to efficiently represent and code the various regions of the scene as well as the motion information. The algorithm involves the following steps: Projection (time tracking of the regions), Partition and Motion coding, Synthesis, Modification of the mesh topology (introduction of new regions and elimination of useless regions) and texture coding.