VIDEO CODING USING NON-MANIFOLD MESH

Nathalie Cammas[1], *Stéphane Pateux*[1], *and Luce Morin*[2]

[1]France Télécom R&D, 4 Rue du Clos Courtel, 35510 Cesson-Sévigné, France
[2] IRISA, Campus de Beaulieu, 35042 Rennes, France
phone: + (33) 2 99 12 49 36, email: nathalie.cammas@francetelecom.com

ABSTRACT

In this paper, we present a novel mesh structure for representing the motion in occlusion areas for video analysis and video coding purposes.
We use the notion of crackline which represents a rupture in a motion field and we create a non-manifold mesh that allows to estimate the motion in the occlusion areas without degeneration of the triangles. The non-manifold mesh can adequately represent the discontinuities in the motion introduced by moving objects and it improves the motion estimation in the occlusion areas.
Results show the better prediction of the texture in those areas, thus improving the coding efficiency.

1. INTRODUCTION

Mesh-based motion analysis, and more generally mesh-based image and video representation have given rise to more and more research efforts over the few last years [1, 2, 3, 4]. In the specific case of video coding, mesh-based motion representation has many advantages such as smooth displacement field preventing blocking artifacts at low bitrate and compact field description thanks to a low number of nodes. Mesh-based motion representation provides more precise and complex motion representation than block-based approaches.
A mesh-based model provides a continuous representation of motion in a scene. However, motion is not continuous. In a natural scene, objects appear, disappear, cross each other, rotate, and thus create occlusions that can not be easily represented using a continuous mesh.
Motion estimation in occlusion areas is difficult and creates stretched or compressed patches involving poor quality prediction. To constrain the patches deformation, some techniques force the mesh nodes displacements using constrained regions [5, 6]. Other techniques [7, 8, 9] apply a post-processing step to correct the nodes that have created a degeneration. These techniques allow to prevent patches degeneration in the motion estimation but introduce false motion in occlusion areas as they force a smooth motion in an area where the real motion is not smooth.
Adaptive meshes in occlusion areas are proposed in [8]. Remeshing is performed in the occlusion area to adapt the mesh structure to scene contents. But, the drawbacks are that the technique needs constant successive structure modification and the structure is not coherent along the video sequence.
In this paper, we propose a new local adaptive mesh structure based on the notion of cracklines [10, 11] which provides discontinuous motion representation in occlusion areas and which is suited for long term texture tracking.
Section 2 introduces the notion of cracklines, section 3 details the mesh structure and the motion estimation, section 4 presents our results and section 5 concludes this paper.

2. THE NOTION OF CRACKLINES

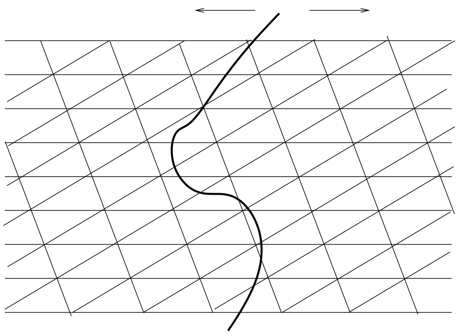
The notion of crackline has been introduced in [11]. A crackline represents a rupture or a discontinuity in the motion field. It is positioned on the contours of the objects that have created the occlusion.
The notion of crackline is illustrated by the figure 1. The mesh is broken along the crackline so that the meshes from both sides of the crackline can move independently with respect to each other. The mesh is then propagated from each side, creating overlapped triangles. This technique allows to avoid elongated triangles and thus provides a better estimation of the motion in the occlusion areas.
A crackline is defined by the contour of an object. It can be a close contour of an object or an open contour. The case of the close contour of an object is close to an object-based representation. The case of an open contour necessitates a particular handling in the remeshing process at the crackline extreme points.
The mesh is broken along the cracklines but only in the occlusion areas. Outside the occlusion areas, motion is continuous and it is represented by the classical 2D mesh. Inside the occlusion zones and along the cracklines, the motion is discontinuous and the representation is similar to an object-based representation. In these areas, the mesh is locally divided in two parts which are joined at the crackline extreme points. At these points, the mesh becomes non-manifold.

3. MOTION ESTIMATION WITH CRACKLINES

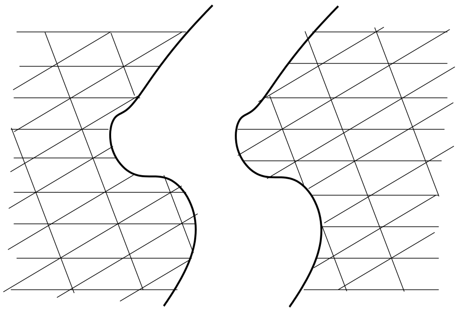
This section presents an overview of the motion estimation scheme using cracklines, the construction of the non-manifold mesh and the reconstruction of a frame using the non-manifold mesh.

3.1 Overview of the motion estimation scheme

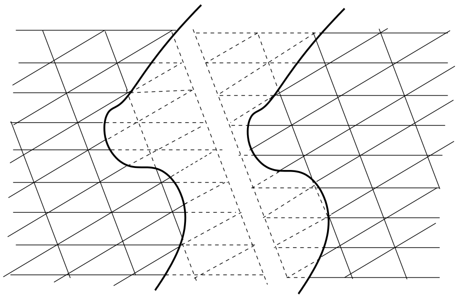
The figure 2 shows an overview of the motion estimation scheme.
A first motion estimation using a uniform triangular 2D mesh is performed on a group of frames of the video sequence. The motion estimation is done through a long-term texture tracking over the group of frames.
Then, from the nodes displacements, a deformation criterion based on the nodes displacement is calculated for each triangle of the mesh and for each frame. This criterion indicates if a triangle is degenerated or not. The occlusion area is then defined on the first frame of the group by the union of degenerated triangles. Given a local object segmentation of the oc-



(a) A crackline in the motion field



(b) Rupture of the mesh



(c) Remeshing each area apart of the rupture

Figure 1: Mesh and crackline

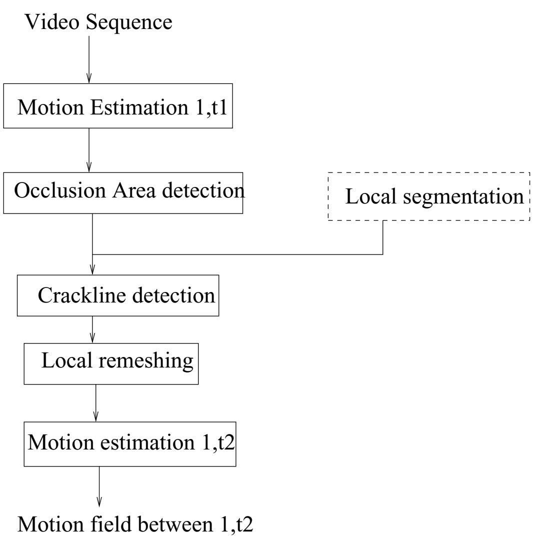


Figure 2: The motion estimation scheme

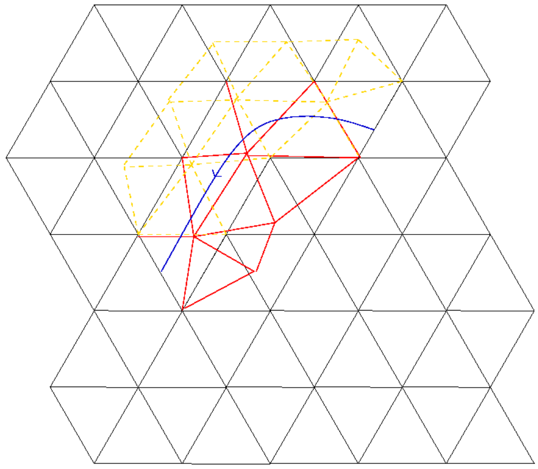


Figure 3: The non-manifold mesh.

clusion area, the crackline is extracted in the occlusion area. Knowing the occlusion area and the position of the crackline, the non-manifold mesh can then be constructed on the first frame of the group and a new motion estimation is performed on the group of frames using the non-manifold mesh.

3.2 The non-manifold mesh construction

The classical 2D mesh is first broken along the crackline. Then, each side is remeshed using edges of triangles from the original mesh to create new triangles. The remeshing is propagated outside the occlusion area to take into account appearing texture during motion estimation. As the mesh is hierarchical, the remeshing is propagated in the hierarchy until the disappearance of the crackline at a coarse level. The figure 3 shows the resulting non-manifold mesh at a fine level. The dark blue line is the crackline. The black lines represent the original 2D mesh. Red and yellow lines represent the remeshed parts from each side of the crackline: the red lines are on the front, the yellow lines are on the background. On this figure, we can see that the mesh is non-manifold at the extreme points of the crackline. At these points, some edges share more than two triangles.

3.3 Frame reconstruction

The remeshing process has created overlapping triangles. This implies that when reconstructing an image using the non-manifold mesh, a pixel may have several candidate luminance values depending on the triangles that can reconstruct this pixel. To select the correct value among the multiple candidate values for one pixel, we use a z-order technique coupled with visibility masks defined for each triangle to indicate which triangles are visible at time t.
The z-order technique is similar to the z-buffer in 3D. Each node carries a z-value to indicate if the triangle is on the front or on the background. As the crackline can be anywhere on a triangle, visibility masks are defined for each triangle using the position of the crackline. Visibility masks indicate which part of the triangle is visible. To avoid the apparition of unknown samples in the case of lossy coding of the crackline, only the front triangles carry the crackline and visibility masks are only calculated for those triangles. The figure 4 illustrates the z-order technique and the visibility masks.

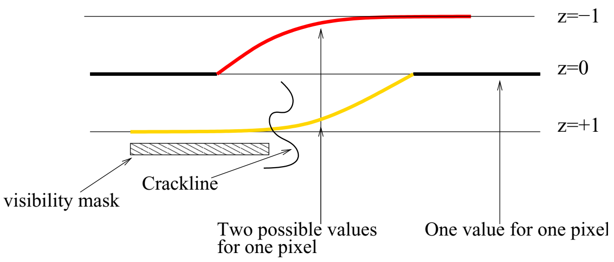


Figure 4: The z-order technique and the visibility masks. This figure represents a cut of the mesh from the figure 3

The z-order technique coupled with visibility masks defined for the triangles allows to reconstruct an image with no ambiguity about the luminance values used.

4. RESULTS

This section presents the motion estimation results obtained with the non-manifold mesh and preliminary video coding results.

4.1 Improvement of the motion estimation

The motion estimation scheme has been presented in section 2. Motion estimation has been performed on groups of 8 frames. More details about mesh-based motion estimator can be found in [10].

In our experiments, we use a priori known segmentation masks to detect the crackline in the occlusion area. We use the sequence *Mobile And Calendar* at CIF, 30Hz. In the *Mobile And Calendar* sequence, the occlusion is positionned between the red ball and the calendar. The triangles belonging to these two objects elongate when the ball moves away from the calendar.

The figure 5 shows the motion estimation results obtained using a manifold mesh and our non-manifold mesh. For non-manifold mesh, the green-blue line shows the crackline, red and dark blue lines represent the discontinuous mesh, locally separated along the crackline. We can see that using the non-manifold mesh, the triangles are no more elongated in the occlusion areas.

The figure 5 also shows the prediction of the texture at frame 8 estimated from frame 1. The prediction is defined as the frame 1 which has been motion-compensated on the frame 8. We can see that the non-manifold mesh allows to better predict the texture in the occlusion area, namely the figures "'26,27,28"' in the calendar.

4.2 Video coding preliminary results

We use the non-manifold mesh for representing the motion in a video coding scheme. The video coding scheme used is presented in [12]. The scheme is based on an analysis-synthesis approach. The analysis performs motion estimation using the non-manifold mesh and extracts the texture information by mapping the frames from the video sequence on a reference sampling grid. The motion and the texture are then coded separately using t+2D wavelet transform and progressive encoding. At the decoder side, the synthesis step warps the texture frames using the decoded motion field to reconstruct the video sequence.

With a manifold mesh, the coding scheme has to code two

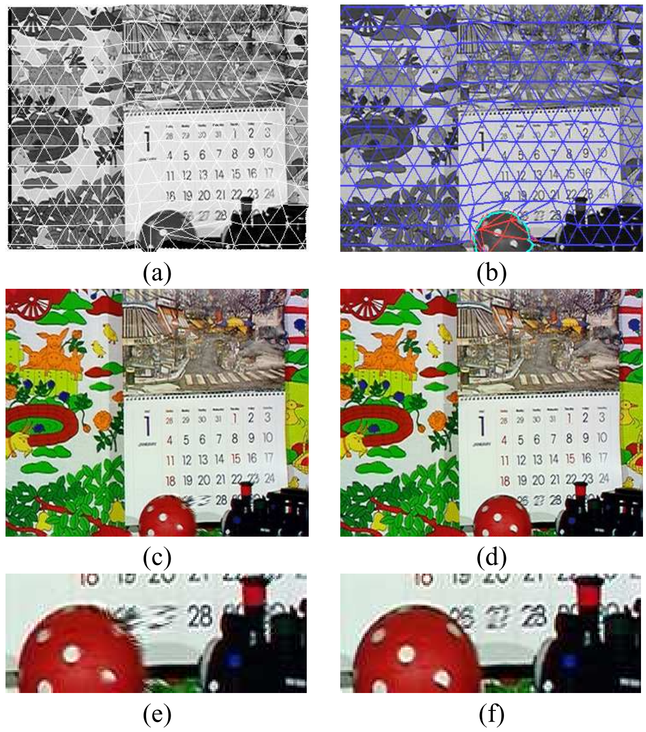


Figure 5: Motion estimation results: comparison between manifold and non-manifold mesh. (a) Motion estimation with manifold mesh, (b) motion estimation with non-manifold mesh, (c) texture prediction with manifold mesh and (e) zoom in the occlusion area, (d) texture prediction with non-manifold mesh and (f) zoom in the occlusion area.

	MM	Non-MM	Crackline
Mobile And Calendar	8.1kb/s	7.9kb/s	1kb/s
Erik	18.68kb/s	18.20kb/s	3kb/s
Flower Garden	20.70kb/s	17kb/s	2kb/s

Table 1: Coding costs of motion and crackline for manifold (MM) and non-manifold (Non-MM) meshes

kinds of information for reconstructing the video. They are the motion (that is the nodes displacement) and the texture frames. When using a non-manifold mesh, a third information must be coded which is the crackline. In our scheme, we use a scalable contours coding scheme presented in [13].

The table 1 shows the coding costs of the motion and the crackline for the manifold mesh and the non-manifold mesh for the three sequences *Mobile And Calendar*, *Erik* and *Flower Garden* at CIF, 30Hz. We can see that the use of the non-manifold mesh allows to reduce the coding cost of the motion information since the motion has been coded at the same quality for both meshes. The reduction of the motion cost sometimes allows to integrate the overcost of the crackline in the case of the non-manifold mesh. This overcost is anyway very small.

The figure 6 shows the reconstructed sequences with the texture decoded at 512kb/s. We can see again that compared with the manifold mesh the non-manifold mesh allows to improve the reconstruction quality in the occlusion area. However, the coding of the texture raises some questions about the

(a) Manifold mesh

(b) Non-manifold mesh

Figure 6: Reconstructed frames

representation of the texture when using non-manifold mesh. With non-manifold mesh, the texture can not be registered as a 2D image, it has to be considered as a 3D surface, because of the overlapping triangles. For our experiments, we choose a multi-layer representation for representing the texture but this representation is not compact and it is redundant. It thus impacts on the coding efficiency of the texture. Future works will deal with finding a novel compact representation of the texture.

5. CONCLUSION

In this paper, we have presented a non-manifold mesh structure using the notion of cracklines. The non-manifold mesh allows to represent the discontinuities introduced in the motion field by moving objects. The mesh has been used in a motion estimation scheme and we have shown the improvements of the motion estimation and the texture prediction in occlusion areas when using the non-manifold mesh. The presented structure is a general structure and it can be used in any video analysis applications which necessitate motion estimation.
The non-manifold mesh has been integrated in a video coding scheme. It has allowed to reduce the motion coding costs and has shown encouraging preliminary results for texture coding. We have seen that the non-manifold mesh raises the question of the texture representation as a 3D surface and its coding. Future works will address this problem.

REFERENCES

[1] Y. Nakaya and H. Harashima, "Iterative motion estimation method using triangular patches for motion compensation" *Proceedings of SPIE Conference on Visual Communication and Image Processing*, Boston, USA, Nov 1991, vol 1605-2, pp. 500-510.

[2] Y. Wang and O. Lee, "Active mesh-A feature seeking and tracking image sequence representation scheme" *IEEE Transactions on image processing*, vol 3:5, pp. 610-622, Sept 1994.

[3] C. Toklu and A.T. Erdem and M.I. Sezan and A.M. Tekalp, "Tracking motion and intensity variations using hierarchical 2-D mesh modeling for synthetic object transfiguration" *Graphical models and image processing*, number 0046, pp. 553-573, 1996.

[4] M. Dudon and O. Avaro and C. Roux, "Triangular active mesh for motion estimation" *Signal Processing: Image Communication*, vol 10:21, number 41, 1997.

[5] Y. Wang and O. Lee and A. Vetro, "Use of two-dimensional deformable mesh structures for video coding, part II- The analysis problem and a region-based coder employing an active mesh representation" *IEEE Transactions on circuits and systems for video technology*, vol 6:6, pp. 647-659, Dec 1996.

[6] N. Laurent and P. Lechat and H. Sanson, "Limitation of triangles overlapping in mesh-based motion estimation using augmented lagrangian" *5th IEEE International Conference on Image Processing, ICIP'98*, Chicago, USA, Oct 1998, vol 2, pp. 223-227.

[7] C. Toklu and A. M. Tekalp and T. Erdem, "2-D Triangular mesh-based mosaicking for object tracking in the presence of occlusion" *Proc. of SPIE Visual Communication and Image Processing, VCIP'97*, pp. 328-337, 1997.

[8] Y. Altunbasak and A.M. Tekalp, "Occlusion-adaptive, content-based mesh design and forward tracking", *IEEE Transactions on image processing*, vol 6:9, pp. 1270-1280, Sept 1997.

[9] N. Laurent, "Hierarchical mesh-based global motion estimation, including occlusion area detection" *7th IEEE International Conference on Image Processing, ICIP'00*, Sept 2000, Vancouver, Canada.

[10] S. Pateux and G. Marquant and D. Chavira-Martinez, "Object mosaicking via meshes and cracklines technique. Application to low bit-rate video coding" *Proceedings of Picture Coding Symposium 2001*, Seoul, Korea, 2001.

[11] G. Marquant and S. Pateux and C. Labit, "Mesh and "Crack Lines": Application to object-based motion estimation and higher scalability" *7th IEEE International Conference on Image Processing, ICIP'00*, Sept 2000, Vancouver, Canada.

[12] N. Cammas and S. Pateux, "Fine grain scalable video coding using 3D wavelets and active meshes" *SPIE Visual Communications and Image Processing, VCIP 2003* Santa Clara, USA, Jan 2003.

[13] M. Chaumont and N. Cammas and S. Pateux, "Fully scalable object based video coder based on analysis-synthesis scheme" *IEEE International Conference on Image Processing, ICIP'03* Barcelona, Spain, Oct 2003.