# Partition projection in videos by global and local block-matching

Guillaume Foret, Pascal Bertolino, David Cibaud

# PARTITION PROJECTION IN VIDEOS BY GLOBAL AND LOCAL BLOCK-MATCHING

*Guillaume Foret, Pascal Bertolino, David Cibaud*

Laboratoire des Images et des Signaux,
BP 46, 38402 Saint Martin d'Hères, France
Guillaume.Foret, Pascal.Bertolino@lis.inpg.fr

## ABSTRACT

This paper[1] presents an original partition projection method for video sequence segmentation. Let us consider that the partition P(t) of the frame I(t) is already available. The aim of the method is to detect the local changes between the frames I(t) and I(t+1). These changes are processed as uncertainty areas when initializing P(t+1). These areas are segmented and resticked to the unchanged parts to give the final partition P(t+1). The detection of uncertainty areas is achieved by two block-matching processes. First a global block-matching is used to detect drastic changes between two successive frames. Then a local block-matching algorithm applied on specific blocks centered on the partition borders allows to declare uncertainty pixels in the neighbourhood of the new borders. The local segmentation of uncertainty areas is facilitated by using the irregular pyramid segmentation based on an adjacency graphs representation.

## 1. INTRODUCTION

The problem of temporal coherence is widely studied in video segmentation techniques [2] [7] [8]. The reason is that many applications really need to track objects. Once the partition of the first frame is available, all these methods achieve a motion estimation between the current frame and the next one to generate the next partition. The motion information can be used to transmit markers in a morphological watershed algorithm [8] or to predict the initial contour of a video object [2] [7]. In our approach, we use motion estimation to define precisely changes within the partition in order to update the partition locally, from frame to frame. The block-matching algorithm is a good compromise between the computational cost and the quality of measures for this task. That is why we use jointly two complementary block-matching processes to detect which are the areas of the projected partition to update (let us call them uncertainty areas) and which are the parts of the partition that will remain unchanged. The pixels of the uncertainty areas will be resticked to the unchanged parts of the partition with the irregular pyramid segmentation.

Section 2 describes how the irregular pyramid works to provide the initial partition. We will see that its graph representation can be useful as well for local segmentation in the next frames. The two different uses of block-matching to detect changes are presented in section 3. Section 4 is devoted to the building of the new partition. We will discuss about some results in section 5.

## 2. SPATIAL SEGMENTATION

The irregular pyramid segmentation is applied on the first frame of the sequence to obtain the initial partition. This partition could also take into account the hierarchical decomposition of the original image to extract the best partition according to the user's need. Although the first partition could be obtained with any other segmentation technique, even by hand, we shortly describe the principle of the irregular pyramid segmentation since we will also use it later to segment some parts of the images.

The irregular pyramid [5] is a region growing segmentation technique for still images. Its data structure is dedicated to its particularities: each level of the pyramid is represented both with a graph and a partition (fig 1.a and b). Once built, the pyramid is a stack of partitions of the original image, and a given level or a combination of several levels is a proper segmentation at a given resolution [1].

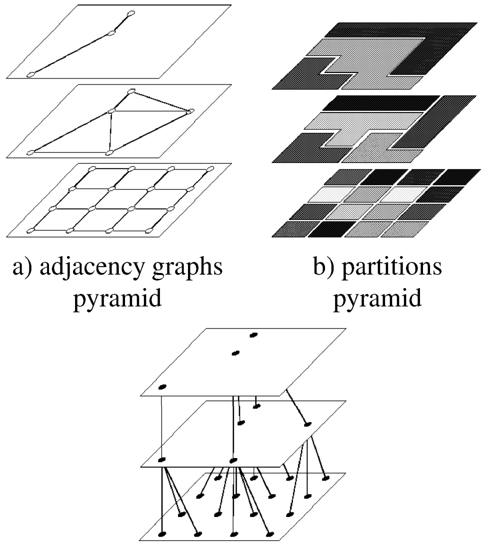
At the beginning of the construction, an adjacency graph (vertices linked with edges) is obtained using the pixels of the original image as initial vertices. Then the following steps correspond to the construction of a new level:

From the adjacency graph a *similarity graph* is built, linking each pair of similar adjacent vertices.

A local *decimation* process performed on the similarity graph permits to locally choose the vertices of the next level (survivors). They will be the regions of the next level.

In the *grouping* phase, non-surviving vertices choose in their neighbourhood the most similar surviving vertex to be linked to, in order to form bigger regions in the next level through sons-to-father links (fig 1.c).

[1] This work was supported by the French RNRT-OSIAM project



a) adjacency graphs pyramid

b) partitions pyramid

c) hierarchical structure (sons-to-father links)

**Fig. 1**. The three structures defining the irregular pyramid (with a 4 x 4 image sample)

For this new level, an *adjacency graph* is redefined to fit the neighbourhood of the new graph.

This new level is now available to be processed in its turn to provide the next level and so on, until the apex of the pyramid is reached when no more fusion can be performed.

## 3. PROJECTING THE CURRENT PARTITION

Our projection method is based on the well known block-matching algorithm used in many video coding applications. We recall briefly how it works.

### 3.1. Prediction by block-matching

Let us consider two successive frames I(t) and I(t+1). We can build a prediction of I(t+1) from I(t) with the block-matching algorithm. There are two possibilities: the backward mode and the forward mode.

In the backward mode, the algorithm involves the division of I(t+1) into small square blocks. For each block of pixels, a search is conducted within a confined window in I(t) to locate the best matching block. Then a prediction of I(t+1) is approached by replacing each block by the corresponding one coming from I(t). Some differences (called prediction errors) can occur between I(t+1) and its prediction.

In the forward mode, the division into regular square blocks is applied to I(t). A motion vector is calculated for each source block of I(t) by searching the best matching block in I(t+1). Then a prediction of I(t+1) is given by projecting each block according to its motion vector. In this case, the predicted image contains uncovered areas either because of the overlapping between projected blocks or because of drastic changes between I(t) and I(t+1) that prevent from matching one or more blocks.

In the following we consider only the forward mode. We use the Block Sum Pyramid Algorithm (BSPA) [3]. It is based on a fast motion estimation method called the successive elimination algorithm (SEA) [4], which can achieve the same estimation accuracy as the Full Search Algorithm (FSA) while requiring less computation time. Small blocks are used in order to have a more accurate estimation (8x8 pixels for QCIF format videos). When performing the block-matching, a similarity threshold is used to avoid attributing a wrong motion vector to a block.

In the following two subsections, we present how the block-matching method is used, first to detect uncovered areas and then to predict the position of the borders in P(t+1).

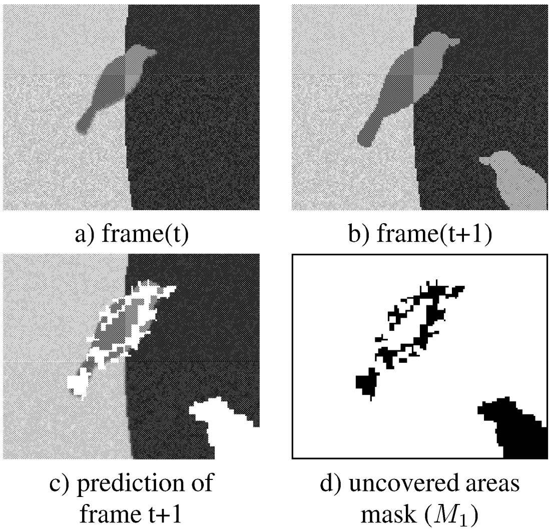
### 3.2. Global block-matching: Uncovered areas detection

The global block-matching used is a classical block-matching in forward mode applied on the whole image I(t). Firstly the motion between I(t) and I(t+1) is estimated. Then a prediction of I(t+1) is achieved by projecting all the blocks of the current frame. As we have seen in the previous paragraph, this predicted image may contain uncovered areas. Most of these areas are due to motion estimation errors or to the modification of the scene content (appearance / disappearance of an object). Indeed when a new object appears, it can't be predicted from the content of the current frame. Figure 2 illustrates this process: two successive frames (fig 2.a and b) show the case of a scale change with a new object appearance. Figure 2.c gives the prediction of I(t+1) where pixels of the uncovered areas remain white. Let $M_1$ be the mask of these uncovered areas (fig 2.d).

### 3.3. Local block-matching: Border projection

To achieve a partition projection, the motion vectors found between I(t) and I(t+1) are applied to the partition of the segmented image P(t) (fig 3.b). We notice that this projection is not accurate close to the contour of the regions. We could expect this result because the matching of blocks containing contours is more difficult. To affine this estimation the authors in [6] propose to subdivide non homogeneous blocks.

In our approach we perform locally a second block-matching between I(t) and I(t+1), taking into account only specific source blocks. These blocks are accurately disposed in I(t), at places where region borders are in P(t) (fig 3.c). The block-matching algorithm is then performed between I(t) and I(t+1) on these blocks. This method improves both



a) frame(t)

b) frame(t+1)

c) prediction of frame t+1

d) uncovered areas mask ($M_1$)

**Fig. 2**. Uncovered areas detection

the number of matches and their quality. The motion vectors obtained with this second block-matching are applied to the corresponding blocks of the partition P(t) to provide another projection. An example of a projected partition using only blocks centered on the borders is presented in figure 3.d. If we compare this projected partition to the one presented in figure 3.b, we can notice an improvement of the projection quality.

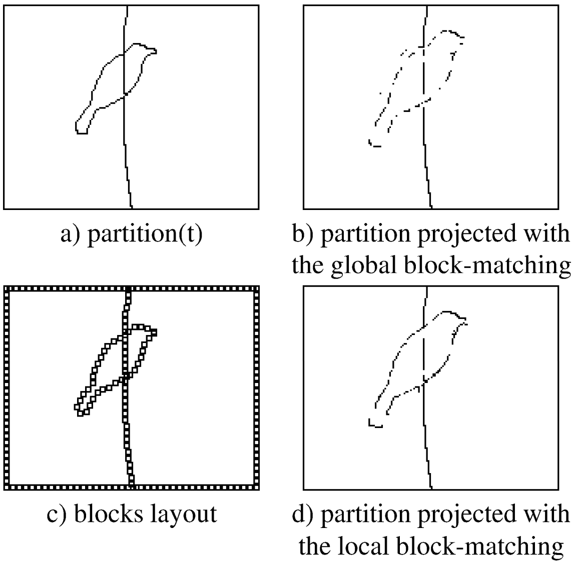
## 4. BUILDING THE NEXT PARTITION

Using the current partition P(t), we build P(t+1) by segmenting I(t+1) locally only where it is necessary: We define some uncertainty areas in I(t+1) that we need to segment at the pixel level. This section presents how the next partition is prepared and updated.

### 4.1. Preparing the next partition

To prepare P(t+1) we build an uncertainty area mask combining two masks obtained through the two block-matching algorithms. The first one, $M_1$, is described in paragraph 3.2 contains uncovered areas (fig 2.d). It is necessary to segment these areas to take into account the appearance of new object in the partition.

Even if the region borders have been correctly predicted during the local block-matching, it is important to refine their exact localisation. So the pixels of the projected border blocks need to be resegmented. Let $M_2$ be the set of theses pixels. Figure 4.a shows such a mask. Nevertheless, as we can see on this example, the above mentionned blocks don't form closed contours. It happens when strong object deformation or zoom-in occur. That is why, a connecting process must be applied to the projected blocks: The connection of two successive blocks is obtained by filling the smallest bounding box that contains these two blocks. We have then a closed contour for each region of the projected partition (fig 4.b).



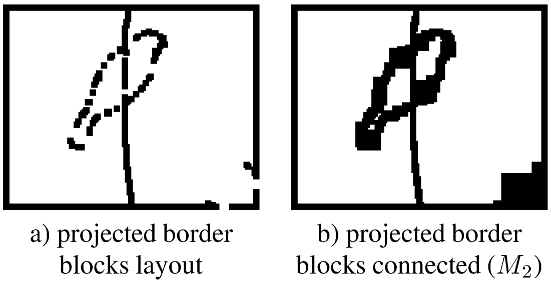
a) partition(t)

b) partition projected with the global block-matching

c) blocks layout

d) partition projected with the local block-matching

**Fig. 3**. Local block-matching for partition projection



a) projected border blocks layout

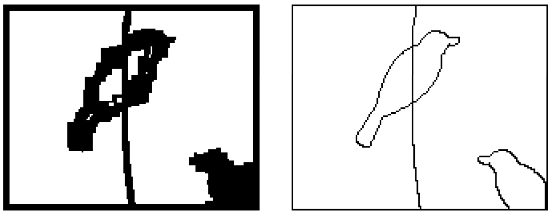
b) projected border blocks connected ($M_2$)

**Fig. 4**. Detection of areas around projected borders

Then the two masks $M_1$ and $M_2$ are combined with a logical $OR$ to provide a unique mask of the uncertainty areas $M$ (fig 5.a).

### 4.2. Updating the next partition

The final mask $M$ of the uncertainty areas (fig 5.a) is applied on the next original frame I(t+1) to build a partition. In this partition, each uncertainty pixel takes the value of the original pixel, whereas the connected components of unchanged pixels take the mean value of the corresponding original pixels. An adjacency graph gathering the independent pixels and the existing regions is initialized. Attributes (area, standard deviation...) are computed for each vertex of this graph. Then this adjacency graph can be processed through



a) uncertainty areas mask ($M$)

b) partition(t+1)

**Fig. 5**. Updating the next partition

the different steps of the pyramid segmentation (section 2) in order to merge progressively the uncertainty pixels to the already segmented regions. So the extension of the labels into the new frame is warrant. Besides we have to notice that the pyramid model allows the creation of new regions from uncertainty pixels (new object segmentation). The updated partition of our example is presented in figure 5.b.

## 5. RESULTS

At the moment, we have limited our tests on simple sequences to check the good evolution of the algorithm in particular cases. The local block-matching algorithm allows to track objects that undergo a local deformation or a zoom-in effect. Rotation prediction which is a weakness of block-matching is managed by the global block-matching that forces a segmentation of the border areas where the matching fails.

We have applied this algorithm on real sequences, but we need to simplify the partition obtained with the irregular pyramid before projecting it. With a complex partition, too much pixels are declared uncertain and our partition projection becomes useless. That is why we plan to achieve an interface allowing partition simplification by the user. In the same way, if the motion present in the sequence is too strong, uncertainty areas are too large in the projected partition. So we limited our study to sequences with low motion.

## 6. CONCLUSION

We have presented a preliminary approach on segmenting image sequences with the irregular pyramid model: For that we have developed a partition projection method based on a double block-matching. The projection gives a coarse partition of the next frame, which is the first level of the irregular pyramid. Since tests on simple partitions give convincing results, we plan to apply our method on more complex sequences. This application will be used in the context of user-assisted segmentation, indeed the user could simplify the first partition to obtain more useful results. Besides we notice that the multiresolution aspect of the pyramid can be taken into account during the user interaction to select the best resolution of semantic video object.

## 7. REFERENCES

[1] P. Bertolino and A. Montanvert. Multiresolution segmentation using the irregular pyramid. In *IEEE International Conference on Image Processing, ICIP'96*, pages 257–260, Lausanne, Switzerland, 1996.

[2] C. Gu and M.C. Lee. Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, September 1998.

[3] C.H. Lee and L.H. Chen. A fast motion algorithm based on the block sum pyramid. *IEEE Transactions on Image Processing*, 6(11), November 1997.

[4] C.W. Lin, Y.J. Chang, and Y.C. Chen. Hierarchical motion estimation algorithm based on pyramidal successive elimination. In *Published on International Computer Symposium*, 1998.

[5] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, April 1991.

[6] M. Pardas and P. Salembier. *Joint Region and Motion Estimation with Morphological Tools*, pages 93–100. Mathematical Morphology and Its Applications to Image Processing - J. Serra and P. Soille - Kluwer Academic Publishers, 1994.

[7] D.K. Park, H.S. Yoon, and C.S. Won. Fast object tracking in digital video. *IEEE Transactions on Consumer Electronics*, 46(3):785–790, August 2000.

[8] P. Salembier, F. Marques, M. Pardas, R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):60–73, February 1997.