Moving Object Extraction with a Localized Pyramid

Jérémy Huart, Guillaume Foret, Pascal Bertolino

Moving Object Extraction with a Localized Pyramid

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

Abstract

In this paper, we present a tracking initialization method that combines a rough extraction of moving objects and a refined segmentation of their contours. The extraction of moving objects is obtained with a classical global motion compensation. To obtain accurate contours of the objects, a spatial segmentation is performed with an original localized graph pyramid that focuses the segmentation process either on the object areas or on their borders.

1. Introduction

Many research works on object tracking within video shots have been (and are still) conducted. In numerous approaches, tracking works well if the initialization of the entities to be tracked has been performed carefully, that is if each entity mask is accurately located on the true contour of the entity. This accuracy can seldom be achieved manually. Moreover, accurate manual interaction is not well felt by the user since it is both time consuming and boring. On the other hand, objects of interest often have a motion that is different from the global motion of the camera and may be roughly extracted [6]. The proposed method offers a general framework for the initialization of a tracking process as well as an effective implementation. In the following, we present the principle of the localized pyramid used to perform spatial segmentation within Regions Of Interest, then how these ROIs are initially obtained. At last, results illustrate the potential of this approach. The functional block diagram of the method is shown in figure 1.

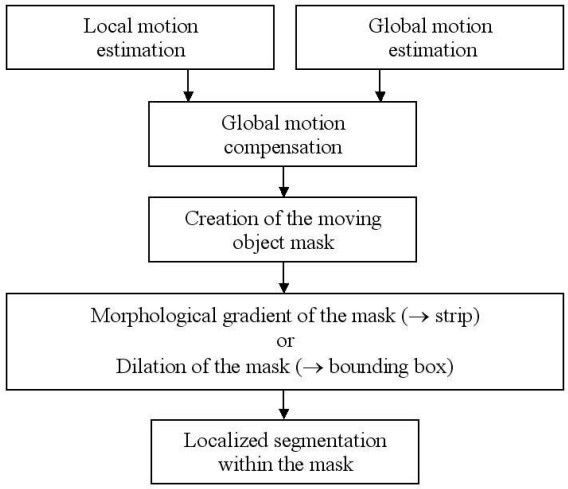


Figure 1. Block diagram of the method

2. The localized pyramid

2.1. Main principles

The graph pyramid (aka irregular pyramid) [5] is a powerful tool that provides multiresolution segmentations during a single process. The principle of this method is to initialize an adjacency graph, where every vertex corresponds to a pixel of the image. Using a local (i.e. pixel independent) algorithm performed on the whole image, similar neighboring pixels can merge, yielding a decreasing number of vertices that actually stand for regions. Regions i and j are similar if their average YUV color distance is lower than a given threshold: $d(YUV(R_i), YUV(R_j)) < T$. The process is iteratively performed on successive graphs until no more fusions is possible.

Usually, the graph pyramid is initialized with as many vertices as the number of pixels in the image, in order to perform the segmentation of the whole image (figure 2). In a localized pyramid, only a subset of the image pixels are associated to vertices, while the rest of them is associated to one (or a few number of) root vertex (figure 3). Root vertices are large regions that will belong to the final segmentation (as the background for instance). Localized segmentation is interesting since it provides faster processing times (only a part of the image is processed) and because the risk of error is spatially limited. Furthermore, the motion information is a fast and easy way to provide the ROIs needed to

build the base of the localized pyramid.

Focusing a particular ROI of the image may be done by different simple means among them: locating the object by coarsely surrounding it (with a bounding box) or coarsely locating its border (with a strip).

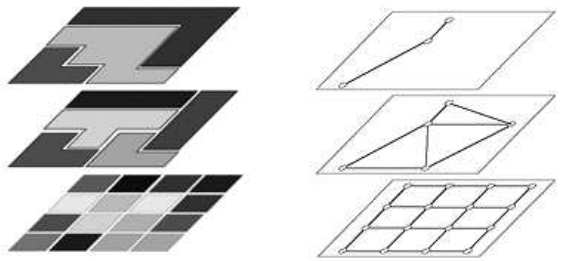


Figure 2. Example of a pyramid built on a 4 $\times$ 4 pixels image: partitions and graphs

Figure 3. Example of a localized pyramid initialization: partition and graph

2.2. Localized pyramid in a bounding box

In this first version, a very common initialization performed with a bounding box is used: a closed shape outside the object to segment is needed. The shape can be for instance a rectangle, an ellipse, or simply a rough mask [4] that contains the whole object (figure 4.a). It is assumed that all the pixels outside the bounding box belong to the background root object (one vertex) while a part of the inside pixels belongs to the object of interest itself. Actually, the pixels inside the bounding box represent an undefined zone. They have to be segmented so that a part of them (the ones similar to the background) merge with the root while the others (the ones of the object) merge together in one or several regions. Such a process not only extracts the object but also provides a segmentation of this object in homogeneous regions (see figures 8.b and 9.b).

(a) Bounding box method (b) Strip method

Figure 4. Examples of ROI initialization

2.3. Localized pyramid in a strip

This approach needs a rough knowledge of the localization of the contour of the object: we make the assumption that the true contour is located within a thick strip (figure 4.b). Using such a strip induces 3 zones: outdoor, indoor and the in-between strip area. Outdoor pixels are assumed to belong to the background object (a first root vertex) while indoor ones are assumed to belong to the object of interest (a second root vertex). The whole lot of pixels forming the strip represent an undefined zone. They have to be segmented so that they merge with one of the two roots. Eventually some of them may merge together in one or several regions without merging with a root. Due to their ability to refine the object borders, localized pyramids in a strip can also be used for accurate object tracking [1].

3. Automatic localization of the ROIs

In order to perform an automatic localized segmentation, the above mentioned bounding box or strip must be automatically positioned in the image. This is achieved with a motion analysis between two frames that contain the objects of interest to extract, as developed in this section. Objects in motion are supposed to have a motion different to the one induced by the camera motion (or global motion).

3.1. Estimation of the local motion

The estimation of the local motion is performed with a fast block-matching algorithm: the *Block Sum Pyramid Algorithm* [2]. This process provides a local estimation of the motion between two consecutive frames $I1$ and $I2$: a motion vector is classically assigned to each of the N square blocks of the frame. In our experiments, 8×8 pixels blocks are used.

3.2. Estimation of the global motion

The parametric model of the global motion is obtained in two phases [3]: first coarsely on the whole image and then

more precisely on the whole image but the moving objects.

A 4 parameters rigid motion is calculated between $I1$ and $I2$ thanks to the Helmert transformation that includes translation (in x and y), rotation and a scale factor as following:

$$\begin{pmatrix} x_i'' \\ y_i'' \end{pmatrix} = \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} a_3 \\ a_4 \end{pmatrix} \quad (1)$$

The couples $(x_i'',\ y_i'')$ et $(x_i,\ y_i)$ represent respectively the central position of block i in the predicted image and in the current image. a_1, a_2, a_3 et a_4 are the parameters value to determine.

The relationship between each block i and its projection by block-matching can be noted:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} a_{1_i} & -a_{2_i} \\ a_{2_i} & a_{1_i} \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} a_{3_i} \\ a_{4_i} \end{pmatrix} \quad (2)$$

The parameters a_1, a_2, a_3 et a_4 have to minimize on the whole N image blocks, the quadratic error Φ between the positions $(x_i',\ y_i')$ estimated by the block-matching and the positions $(a_1x_i - a_2y_i + a_3, a_2x_i + a_1y_i + a_4)$ predicted by the model itself. This cost function is defined by:

$$\Phi = \sum_{i=1}^{N} [(a_1x_i - a_2y_i + a_3 - x_i')^2 + (a_2x_i + a_1y_i + a_4 - y_i')^2] \quad (3)$$

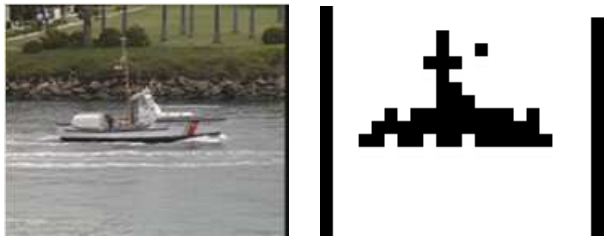
The minimization of the criterion (3) is realized with the least squares by using the singular value decomposition (SVD). The code is available in the *Numerical Recipes* [7].

The thresholded Euclidian distance between the two predictions $(x_i',\ y_i')$ and $(x_i'',\ y_i'')$ then allows to distinguish the blocks that don't have the global motion (figure 5). The whole process can be reiterated (and thus refined) without taking into account these blocks.

3.3. Automatic extraction of the ROIs

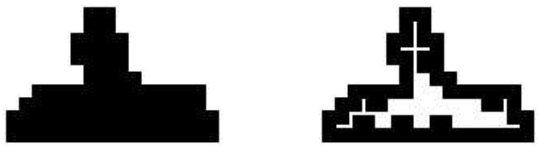
The bounding box is obtained by the dilation of the foreground blocks stemming from the motion analysis (figure 6.a). The strip is the result of the subtraction of the dilated and the eroded of the foreground blocks (figure 6.b). The foreground blocks located at the periphery of the image are rejected in order to avoid occlusions and disocclusions problems due to the camera motion (figure 5.b).

A minimum size threshold or a morphological filtering can be used to avoid too many ROIs in the case of noisy videos. Nevertheless, the graph representation allows as many ROIs as possible, that is, any number of objects sharing the same background.



(a) Original image (b) Temporal binary mask

Figure 5. Example of temporal binary mask extracted by the global motion compensation



(a) Bounding box (b) Strip

Figure 6. ROIs obtained automatically

4. Results

Experiments have been carried out on CIF videos (352×288 pixels). Running a 2 GHz Pentium IV, processing time is between one and two seconds. Motion analysis is very fast and most of the processing time is devoted to segmentation. For the same ROI, the strip is about twice faster than the bounding box since the number of pixels involved in the pyramid construction is smaller. The processing time is related to the number of pixels of the undefined zones.

Figures 7.b, 7.d, 8.a and 9.c show results of the method applied on classical video sequences with different camera motions, different background and non rigid objects. Excepted for figure 9.c, no post-processing is applied. The similarity threshold T (see section 2.1) still has to be tuned by the user, even if in most of the cases, a default value gives good results. An adaptive threshold could be calculated mainly in the case of the strip where the pixel information is rather well concentrated and reliable.

Figures 7.d and 8.a compare (from the same ROI mask) the results obtained with the two approaches. The strip allows a more accurate location of the borders but according to the similarity criterion used, some regions may have difficulties to merge with one of the two roots. Figure 7.d shows in black the regions that could not be confidently classified within the object or in the background.

The bounding box provides binary results (object or

background) but with a lower accuracy. This is due to segmentation that is performed on a larger area.

Obviously, whatever the approach chosen, the result quality depends on the ROI mask precision and on the homogeneity of the object versus the background: even if the objects and the background are poorly contrasted, the method works fine if either the object or the background (or both) are homogeneous according to the similarity criterion.

(a) Original

(b) Object extracted (unclassified regions in black)

(c) Original

(d) Object extracted (unclassified regions in black)

Figure 7. Object extraction with the strip approach

5. Conclusion

The quality of the masks has been successfully tested with an object tracking application [1]. For these applications, it is important that the mask initialization be performed automatically, since the tracking process will use as well quantitative color information that is also used in the localized pyramid. The automatic localization of the bounding box or of the strip can be improved, among others the combination of more than two images for the temporal binary mask and the smoothing of the final bounding box or strip. In the future, we aim at using these methods to generate key object dictionaries to describe the video content or for compression norms such as MPEG-4.

(a) Object extracted

(b) Detail of the regions that form the object

Figure 8. Object extraction with the bounding box approach

(a) Original

(b) Detail of the regions

(c) After some regions removal

Figure 9. Object extraction over a textured background, with the bounding box approach

References

[1] G. Foret and P. Bertolino. Label prediction and local segmentation for accurate video object tracking. In *SPIE VCIP'03*, Lugano, Switzerland, 2003.

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

[3] S. Liu, Z. Yan, J. W. Kim, and C. C. J. Kuo. Global/local motion-compensated frame interpolation for low bitrate video. *Image and Video Communications and Processing*, pages 223–234, 2000.

[4] A. Mahboubi, J. Benois-Pineau, and D. Barba. Joint tracking of polygonal and triangulated meshes of objects in moving sequences with time varying content. In *IEEE ICIP'01*, Thessaloniki, Greece, 2001.

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

[6] A. Tekalp. *Digital Video Processing*. Prentice Hall, Inc, 1996.

[7] C. university Press. Numerical recipes in c: The art of scientific computing. Website. `http://gpiserver.dcom.upv.es/Numerical_Recipes/bookc.html`.