IEEE Signal Processing Society International Conference on Image Processing
September 1996, Lausanne, Switzerland

INCREMENTAL MOTION SEGMENTATION IN LOW TEXTURE

Pedro M. Q. Aguiar [1] *José M. F. Moura* [2]

[1] Instituto de Sistemas e Robótica, Instituto Superior Técnico
Av. Rovisco Pais, P-1096 Lisboa Codex, Portugal
E-mail : aguiar@isr.ist.utl.pt
also affiliated with
[2] Electrical and Computer Eng. Dep., Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213-3890, U.S.A.
E-mail : {aguiar,moura}@ece.cmu.edu

ABSTRACT

The paper studies segmentation of *moving* objects with *low texture* in a *low textured* background. We describe an algorithm that resolves the difficulties associated with other approaches by integrating over time the information in the video sequence. We motivate and demonstrate our approach by building the background and moving object world images, important constructs in **Generative Video** [1].

1. INTRODUCTION

Generative video (GV) [1] is a framework for the analysis and synthesis of video sequences. **GV** reduces video sequences to world images and to ancillary data. The world images are augmented views of the world - background world image - and complete views of moving objects - figure world images. The ancillary data registers the world images, stratifies them, at each time instant, and positions the camera with respect to the layering of world images. The world images and the ancillary data are the **GV** representation, all that is needed to regenerate the original video sequence.

In **GV** the operational units are not the individual images in the original sequence, as in standard methods, but rather the world images. **GV** reduces significantly the computational resources, the storage space, and the transmission bandwidth needed to manipulate video sequences.

A major task in **GV** is to derive from the given video sequence the background world image and the figure world images, one for each independently moving object. This paper introduces an algorithm for segmentation of moving objects when the objects and/or the background have no texture. This is a particularly difficult task where, for example, optical flow based methods fail and do not work properly. We develop a computationally simple and efficient algorithm that builds incrementally templates for the moving objects, their world images, and the background world image. The paper presents the results of our algorithm with segmenting moving cars in a real life video clip.

1.1. Assumptions

We make the following assumptions:

- Background dominance: the moving objects are small when compared with the size of the images in the sequence.
- 2D parallel motions: all motions (translation, scalings, rotations) are parallel to the background. In particular this assumes that the relative depth is small when compared with the distance between the background and the camera, or that the axes of rotations are perpendicular to the background.
- Low noise or low illumination variations.
- Low textured objects and background.

The first three assumptions simplify the problem. The assumption of low noise and low illumination variations is not critical. The background dominance hypothesis simplifies the motion determination algorithm but can be easily generalized. The 2D parallel motion is a compromise between for example the length of the video sequence and the accuracy of the representation. The low textured assumption is a distinguishing feature of our work. With low textures, gradient based methods fail to detect the motion of large regions in the background or moving objects.

To overcome the problem incurred by the low textured objects and/or background, we develop below an algorithm that achieves segmentation from motion by integrating over time the information content of the video sequence. This approach departs significantly from current techniques which attempt to segment the moving objects by processing simply two or three consecutive images.

1.2. Related Work

Motion segmentation for general scenes makes use of optical flow techniques. These methods work well if the different regions in the image have noticeable texture in order to overcome the aperture problem [2]. In general, they are inefficient, attempting to solve the problem with a small number of images.

The work of the first author was partially supported by NATO.

A number of papers combine different techniques for segmentation. See [3] for the integration of motion segmentation with color segmentation. These methods lead to complex and time consuming algorithms.

References [4] and [5] describe one of the few approaches that use temporal integration of the information by averaging the images registered according to the motion of the different objects in the scene. After processing a number of images, each of these integrated images should show only one sharp region corresponding to the tracked object. However, this is not the case, unless the background is textured enough to blur the averaged images.

Finally, several papers in the computer vision literature study motion segmentation and tracking based on image features. These algorithms do not lead to dense representations of the moving objects, a major goal of our work.

1.3. Organization of the Paper

Section 2 introduces the necessary notation. The overall algorithm has two distinct phases. The first one is the initialization of the templates of the moving objects. It is described in section 3. The second phase, detailed in section 4, corresponds to the recursive generation of the world images and to the template updating. Experimental results and conclusions are in sections 5 and 6.

2. NOTATION

Capital letters, stand for matrix quantities:

- I_i: frame i in the video sequence.
- B_i: estimate of the background world image, obtained with the first i frames.
- S_i^b: weight matrix needed for the estimation of the background.
- O_i^j: estimate of the world image of moving object j.
- S_i^{oj}: weight matrix needed for the estimation of the world image of object j.
- T_i^j: template of object j.
- U_i^j: "smoothed" version of the template of object j.
- M_{ik}^j: mask of object j, estimated from frames i and k.
- Q_i^j: mask of object j, estimated from the first i frames.
- R_{ik}: detected moving regions between frames i and k.
- D_{Bi}: regions of the image i that differ from the previous estimate of the background world image.

Matrix S_i^b has the dimensions of B_i. The element $S_i^b(x,y)$ is the number of times pixel (x,y) has been observed. The same relation exists between $S_i^{oj}(x,y)$ and $O_i^j(x,y)$.

Each template T_i^j is a binary matrix which defines the region for the moving object j in image I_i. Section 3 describes the method used to initialize these matrices.

Lower case letters represent vectors:

- $p_1, p_2, \ldots, p_n$: estimated positions of the camera relative to the background world image.
- $q_1^j, q_2^j, \ldots, q_n^j$: estimated positions of moving object j relative to the background world image.

Since the main focus of our work is on incrementally generating models by temporal integration and not on motion estimation, we estimate motion with a simple block matching algorithm. Any other method that estimates motions of multiple objects (see [6] for a survey) can be used with our framework. Block motion estimation is accurate enough for our purposes because we deal here with 2D planar motions. To estimate the motion of multiple moving objects, we first estimate the background motion by assuming that it is the dominant motion. After registering the images according to this dominant motion, the motions of the moving objects are estimated by a quad-tree method. We start by defining a binary matrix with ones at the pixel locations associated with all moving objects. This matrix is recursively decomposed into smaller matrices and the motion of each sub-region defined by each sub-matrix is estimated. Then we associate regions with similar motion.

We denote by $A(a)$ the registration of the image matrix A according to the position vector a.

3. TEMPLATE INITIALIZATION

A major problem in segmenting low textured scenes is the initialization of the templates of moving objects. Due to the absence of texture, some regions agree with the motion vectors associated with different objects and/or the background, in the sense that their motion can be described by any of these vectors. In general, it is impossible to decide to what objects these regions belong when only a few frames are taken into account.

To overcome this problem, rather than assuming some form of prior knowledge about the shape of the objects, we integrate over time the information content of the image sequence, in order to get a reliable estimate of the templates of the moving objects.

3.1. Object Mask From a Pair of Images

Given a pair of images, registered according to the estimate of the background motion, $I_i(p_i)$ and $I_k(p_k)$, detect the region whose motion differs from the background motion by:

$$R_{ik} = \begin{cases} 1 & \text{if } |I_i(p_i) - I_k(p_k)| > \eta_1 \\ 0 & \text{otherwise} \end{cases}$$

This region contains the templates of the moving objects in both images $I_i(p_i)$ and $I_k(p_k)$. If there is only one moving object j, and it is textured enough, R_{ik} contains the union of the templates of that object positioned in each of the frames (see example shown in figure 1).

We obtain an estimate of the template of the object j by intersecting R_{ik} with itself, registered according to the estimate of the motion of object j between frames i and k, $M_{ik}^j = R_{ik}R_{ik}((q_i^j - p_i) - (q_k^j - p_k))$. We call the estimate M_{ik}^j a mask of the moving object. In the example of figure 1, this mask successfully detects the moving object.

In the general case, there are several moving objects. We obtain the mask for each one by selecting from $R_{ik}R_{ik}((q_i^j - p_i) - (q_k^j - p_k))$ the region that agrees with

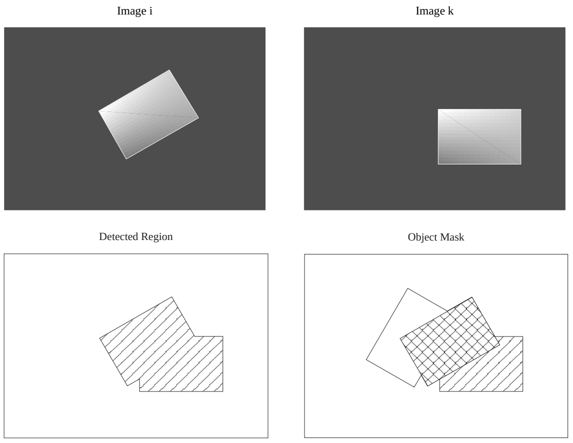

Figure 1: Object mask from a pair of images.

the estimated motion of the object j:

$$M_{ik}^{j} = \begin{cases} R_{ik}R_{ik}((q_i^j - p_i) - (q_k^j - p_k)) \\ \qquad\qquad \text{if } |I_i(q_i^j - p_i) - I_k(q_k^j - p_k)| < \eta_2 \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

3.2. Temporal Integration

When the moving objects contain regions of low texture, R_{ik} does not detect the entire moving region, and M_{ik}^j does not represent a reliable mask of object j. Also, with some particular backgrounds, object shape and motion, or due to noise, M_{ik}^j may contain regions not belonging to the object. To deal with this type of scenes, we take into account masks M_{ik}^j obtained from several pairs of frames i, k, to generate the initial object template.

We compute the average of these masks, after registration. This computation is done recursively by:

$$Q_m^j = \frac{m-2}{m} Q_{m-1}^j + \frac{2}{m(m-1)} \sum_{i=1}^{m-1} M_{im}^j (q_i^j - p_i)$$

We stop this recursion when Q_m^j stabilizes. Finally, the template of the object j is initialized by thresholding Q_m^j:

$$T_m^j = \begin{cases} 1 & \text{if } Q_m^j > \eta_3 \\ 0 & \text{otherwise} \end{cases}$$

In order to incrementally build the templates of the moving objects, we need to track them. This is done by associating with object j the mask M_{im}^j that best matches the averaged mask of previously detected objects Q_{m-1}^j. Every time a mask M_{im}^l does not match any of the masks Q_{m-1}^j, a new object mask is created.

4. WORLD IMAGE GENERATION

After template initialization, we generate the world images. First we process the first m frames, used to estimate the initial templates T_m^j. Then we keep updating the background and objects world images, as well as the templates, with the subsequent frames.

4.1. Background World Image Estimation

Using the template T_m^j, we initialize the background world image, by averaging the first m images in the regions not occluded by the moving objects:

$$B_m = \frac{\sum_i I_i(p_i) \prod_j [1 - T_m^j(q_i^j)]}{\sum_i H(p_i) \prod_j [1 - T_m^j(q_i^j)]}$$

where $H(p_i)$ represents the registration of a constant image $H(x,y) = 1$ according to p_i.

The estimation weights are initialized by:

$$S_m^b = \sum_i H(p_i) \prod_j [1 - T_m^j(q_i^j)]$$

After initialization, the background world image is recursively estimated by averaging the previous estimate with the new image, in the regions not occluded by the moving objects, and taking into account the estimation weights:

$$B_n = \frac{B_{n-1} S_{n-1}^b + I_n(p_n) \prod_j [1 - T_n^j(q_n^j)]}{S_{n-1}^b + H(p_n) \prod_j [1 - T_n^j(q_n^j)]}$$

The weights are updated by:

$$S_n^b = S_{n-1}^b + H(p_n) \prod_j [1 - T_n^j(q_n^j)]$$

4.2. Object World Image Estimation

The initialization of the world image of object j is by averaging the first m frames in the region corresponding to T_m^j:

$$O_m^j = \frac{T_m^j \sum_i I_i(q_i^j - p_i)}{m}$$

The estimation weights are initialized by:

$$S_m^{oj} = m T_m^j$$

The recursive estimate for the object world image is:

$$O_n^j = \frac{O_{n-1}^j S_{n-1}^{oj} + T_n^j I_n(q_n^j - p_n)}{S_{n-1}^{oj} + T_n^j}$$

$$S_n^{oj} = S_{n-1}^{oj} + T_n^j$$

4.3. Template Updating

In order to update the templates of the moving objects, we define a “smoothed” version of them, U_i^j. They are initialized by $U_m^j = T_m^j$. For $i > m$, we compute the crisp template T_i^j by thresholding U_i^j:

$$T_i^j = \begin{cases} 1 & \text{if } U_i^j > \eta_4 \\ 0 & \text{otherwise} \end{cases}$$

To update U_i^j, we start by detecting the regions of the new image that differ from the actual estimate of the background world image:

$$D_{Bn} = \begin{cases} 1 & \text{if } |B_{n-1} - I_n(p_n)| > \eta_5 \\ 0 & \text{otherwise} \end{cases}$$

Then, we increment U_{n-1}^j in the regions of D_{Bn} and T_{n-1}^j that agree with the estimated motion of object j, and decrement it in the regions of T_{n-1}^j not agreeing with that motion:

$$U_n^j(q_n^j) = \frac{n-1}{n} U_{n-1}^j(q_n^j) + \\ +\frac{1}{n}\begin{cases} T_{n-1}^j(q_n^j) & \text{if } |O_{n-1}^j - I_n(q_n^j - p_n)| < \eta_6 \\ D_{Bn} & \text{if } |B_{n-1} - I_n(q_n^j)| < \eta_7 \\ -T_{n-1}^j(q_n^j) & \text{if } |O_{n-1}^j - I_n(q_n^j - p_n)| > \eta_6 \end{cases}$$

Every time D_{Bn} detects regions that do not match any of the templates and do not agree with any of the estimated object motions, a new object is created and its template initialized as described in section 3.

5. EXPERIMENTAL RESULTS

To test the algorithm, we used a sequence of fifteen images obtained from a real outdoor scene. In this sequence, see figure 2, a car moves in front of a panning camera. The car and the background have regions of low texture.

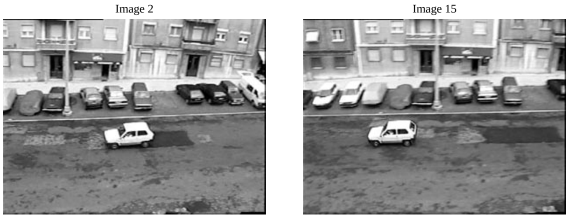

Figure 2: Image sequence. Frames 2 and 15.

Figures 3, 4 and 5 show the results obtained with our method. Figure 3 illustrates the incremental building of the template of the car. The world images of the background and the moving object are presented in figures 4 and 5.

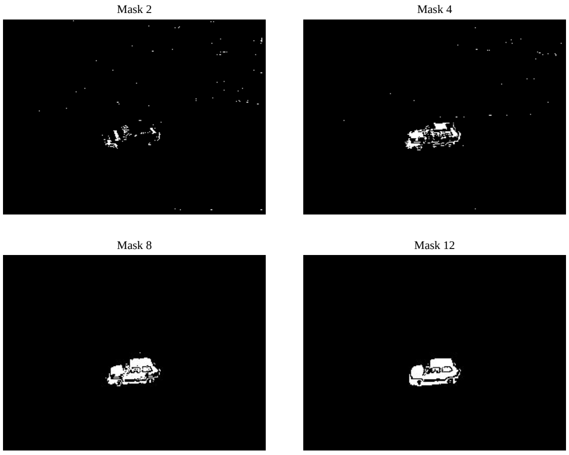

Figure 3: Template of the moving object after processing 2, 4, 8 and 12 images.

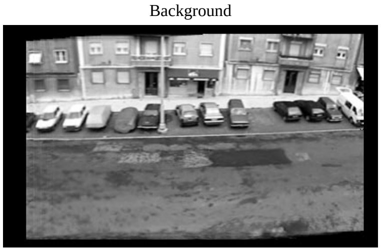

Figure 4: Reconstructed background.

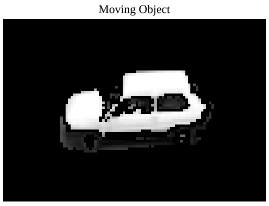

Figure 5: Moving object world image

6. CONCLUSIONS

We have developed a technique to segment an image sequence according to motion attributes. Our approach incrementally builds the templates of the moving objects, their models and the model of the background, by integrating the information of the image sequence over time.

The algorithm described is computationally simple, segmenting moving objects with low texture.

The experimental results we obtained are satisfying and show that our motion segmentation method works well under low textured conditions.

7. REFERENCES

[1] R. S. Jasinschi and J. M. F. Moura. Content-based video sequence representation. In *Proceedings of the IEEE International Conference on Image Processing*, Washington DC, USA, 1995.

[2] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[3] Marie-Pierre Dubuisson and Anil K. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14(1):83–105, 1995.

[4] Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communications ans Image Representation*, 4(4):324–335, December 1993.

[5] Michal Irani, Benny Rousso, and Shmuel Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[6] Serge Ayer. *Sequential and Competitive Methods for Estimation of Multiple Motions.* PhD thesis, École Polytechnique Fédérale de Lausanne, 1995.