Background mosaic from egomotion *

Rémi Megret
École Normale Supérieure de Lyon
46 allée d'Italie
F-69364 Lyon FRANCE
rmegret@ens-lyon.fr

Caterina Saraceno
Starlab NV
Blvd. St.-Michel, 47
B-1040 Brussels BELGIUM
saraceno@starlab.net

Walter Kropatsch
Pattern Recognition and Image Processing group
Vienna University of Technology
Favoritenstr. 9/1832, A-1040 Vienna AUSTRIA
krw@prip.tuwien.ac.at

Abstract

In this paper a framework is presented that produces the mosaic corresponding to the background object of an image sequence. It is based on the dominant motion assumption, which states that the background has a parametric motion and occupates the main part of the images. The foreground objects are localised by their different motion. This localisation is computed together with the background motion in an iterative method. The regions corresponding to the background are then pasted onto the mosaic using classic methods adapted to object elimination or a new mosaicking method based on a striping that takes the foreground objects localisation into account.

1. Introduction

Video sequences generally present a high temporal redundancy, because the background and the foreground objects are repeated over the consecutive images. The mosaicking technique allows to produce a single image that represents a whole shot, by eliminating this temporal redundancy. The general structure of such algorithms involves two steps towards building a static mosaic [3]: first, the images are aligned using a parametric motion model (*registration*), then they are pasted together to produce the mosaic image (*mosaicking*).

When the apparent displacements are well approximated by a simple parametric model, the whole images can be pasted together. This doesn't hold when several distinct motions appear. Based on a segmentation step, the algorithm will choose which part of the image shall go onto the mosaic.

1.1. Alignment of images and motion segmentation

Given an object, finding its motion and its location are two related problems. Indeed, if the general motion is not known, local motion computation is less precise since local contribution to motion field can be small compared to the general motion. In this case, finding the object boundaries is more difficult. Reversely, the motion computation can be biased if the parameters are evaluated on an image that contains no coherent motion or more than one.

We can distinguish two kinds of methods to process registration and segmentation based on motion:

a) The first ones are based on the *preliminary computation of a local motion field*. Regression techniques are then applied to these data to segment images into regions with coherent motions[6, 10], and evaluate their motion.

b) The other methods involve *global alignement based on a parametric model* [3, 7, 8]. When multiple motions are present in the sequence this framework can only be used to find the *dominant motion* under the assumption that the corresponding object occupates the main part of a given region of interest. A multiresolution framework [1] reveals to be well adapted to this problem. It was coupled in [4] with detection of outlier objects to avoid taking them into account, thereby reducing their influence on the computed motion. Because of the expected properties of the background (see section 2) this framework is used for the analysis part of our mosaic building method.

*This work was supported by the Austrian Science Foundation under grants S 7000-MAT and S 7002-MAT.

Whatever alignment method for image pairs is chosen, global alignment frameworks [9, 2] then compose the pairwise parameters to produce aligment parameters between the source images and the mosaic.

1.2. Generation of mosaic

Once images are aligned, they can be pasted onto the mosaic manifold [7], that is a value is computed for each pixel in the mosaic space based on the values of the corresponding pixels from the aligned images.

Mosaicking methods can be classified as combining and partitionning techniques. With combining methods, values are averaged or a median is computed [3]. Sharper mosaics can be obtained by partitioning the mosaic image into regions and copying all the pixels in one region from a same source image; the risk is to produce mosaics with discontinuities at the boundaries of the regions. This partition may be guided by global motion, as in the striping technique [7, 8], or computed from local displacements as in [2].

Object elimination was introduced for combining methods in [10] with the layered representation. In each source image a mask selects the pixels associated with a given layer. The layer mosaic is then produced by combining pixels that lie on the masks and discarding objects situated outside the masks.

Our approach is intended to produce sharp mosaics by using partitionning, while perfoming object elimination.

1.3. Structure of the algorithm

Registration and background segmentation are run in a process involving a few iterations, each one improving the results of the other. This module gives the alignment parameters between pairs of consecutive images and the mask corresponding to the background pixels for each image.

The parameters are then composed to produce global parameters, that describe the alignment between source images and the mosaic manifold.

Using the global parameters images are aligned towards the mosaic coordinates. Then a partitionning of the mosaic is processed out of the aligment parameters and the background masks. Relying on it, pixels are copied from source images to the mosaic. Figure 1 shows a data-flow of this general framework.

2. Registration and Segmentation

In this section we will precise the base framework used to align an image pair, while detecting the moving objects.

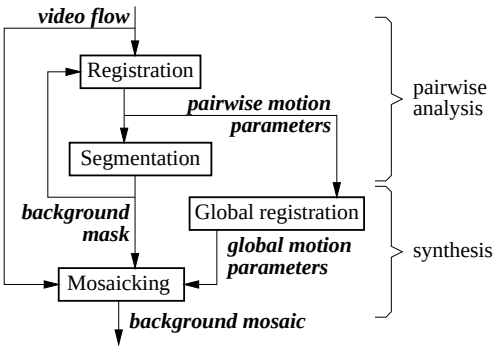

Figure 1. Background mosaic computation

2.1. Framework

We chose to use a dominant motion approach with an associated segmentation as introduced in [4] for the registration of images. Local motion estimation is computed only on aligned images, which increase their accuracy. The background is expected to *occupy the main part of the image*, so the dominant motion effectively represents its motion. As the background is supposed to be *static in the real world* its apparent motion is just the effect of the camera movement. Furthermore it is the farthest object in the image, so a foreground/background segmentation is needed to detect objects that occlude it. To allow the mosaicking, a parametric motion model represents the whole background motion.

For a pair of consecutive images the registration-segmentation process is composed as follows:

1. Initialize the background mask to the mask cue (see subsection 2.2) from the previously registered pair.
2. Repeat steps 3.,4.,5.,6. several times to converge towards the motion and localisation of the background.
3. Find the dominant motion between the two original images using the background mask. Only pixels belonging to this mask are taken into account.
4. Align one image onto the other using the parameters computed in step 3.
5. Segment out foreground pixels.
6. Set the background mask from the result of step 5.

The dominant motion computation is usually processed using a hierarchical method like [1]. The segmentation is based on local motion estimation: pixels with a local motion over some threshold belong to the foreground. This threshold is fixed (typically 1.0 pixel), or decreases at each loop to take into account the non perfect alignment in the first loops.

2.2. Cue for the background mask

In the alignment/segmentation process, each alignment is performed by taking into account only pixels belonging to the background mask. When this mask does not represent the background precisely enough, this introduces a risk that the first steps converge towards the motion of a foreground object instead of the background.

shortened:

We can avoid such an erroneous convergence by using the background mask computed for the previous image pair in the first registration. The only foreground pixels that are then taken into account for the global motion estimation belong to newly occluded regions.

3. Mosaicking

At this stage two kinds of information are available: alignment parameters between each image and the mosaic, and background masks for each source image.

Our purpose is to combine the image pixels to get a mosaic image composed only of background objects. We focus in the following on the computation of the gray value $v(\vec{x})$ for a given pixel $\vec{x}$ on the mosaic.

Let us denote $v_t(\vec{x})$ the gray value of the corresponding pixel in image at time t (this value is computed out by warping the source image according to alignement parameters), and $\mathrm{bg}_t(\vec{x})$ the related value of the background mask ($\mathrm{bg}_t = 1$ for background pixels).

3.1. Classical methods

The simplest method consist in combining these values with a function f like a mean or a median:

$$v = f(v_t \mid t = 1 \dots n)$$

This mosaicking method was extended to take into account the background mask in [10], by combining only pixels corresponding to the background:

$$v = f(v_t \mid bg_t = 1,\ t = 1 \dots n)$$

At the opposite, the striping approach chooses one value among those available. The mosaic image is partitioned into regions, each of which is associated with one source image. It is represented by a partition function $s(\vec{x}) \in \{1 \dots n\}$ that gives the image number from which the pixel value for $\vec{x}$ is to be copied:

$$v = v_{s(\vec{x})}(\vec{x})$$

The striping method may present some discontinuities at the boundaries between regions, although this is limited by the registration procedure. It avoids the blurring caused by the combination of many pixel values, thus producing sharper mosaics.

3.2. Partitioning to discard foreground objects

The point is to define a partition function $s(\vec{x})$ such that only background pixels (as defined by the background mask) are selected. To do so, we recall striping method where image centers are given a higher priority [7]. We extend it to take into account the background mask in the following manner.

Given a pixel $\vec{x}$ of the mosaic, the corresponding pixel in source image number t is associated a positive confidence coefficient $C^t(\vec{x})$ that tells in which measure the pixel should appear in the final mosaic. During combination the most confident pixel is selected:

$$s(\vec{x}) = \mathrm{argmax}_t(C_t(\vec{x}))$$

The striping technique which partitions using a Voronoi tesselation [3] of the centers of source images can be expressed in this model with a confidence coefficient $C^t_{center}(\vec{x})$ decreasing with the distance to the center of images. For example:

$$C^t_{center}(\vec{x}) = \exp(-\|\vec{x} - \vec{x}_{center}\|/\alpha)$$

where x_{center} is the center of the image and α a tuning parameter. This gives higher priority to the center of the source images that are more reliable (see [7]).

Our approach introduces a penalty for pixels near foreground objects. Confidence $C^t_{bg}(\vec{x})$ is zero on foreground mask (the complementary of background mask), and increases with the distance to foreground mask. Denoting $d_{fg}(\vec{x})$ as the distance from $\vec{x}$ to the closest foreground pixel, this can be expressed as follows:

$$C^t_{bg}(\vec{x}) = -\exp(-d_{fg}(\vec{x})/\beta)$$

This kind of function gives the priority to areas far from moving object, thus limiting problems due to unprecise segmentation boundaries.

The final confidence coefficient takes into account those two points of view, by mixing them:

$$C^t(\vec{x}) = \begin{cases} 0 & \text{where } \vec{x} \text{ is not in background mask} \\ C^t_{center}(\vec{x}) + \lambda\, C^t_{bg}(\vec{x}) & \text{elsewhere} \end{cases}$$

An example of such a confidence image is shown in fig.2. White pixels represent foreground pixels, and darker pixels have a higher confidence.

4. Results

The methods presented in this paper were implemented to test them on video sequences. More complete results are described in [5].

suppressed subsection on cue mask test

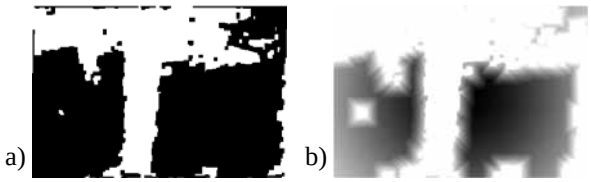

Figure 2. Foreground mask (a) and associated confidence image (b)

Confidence C^t from subsection 3.2 was used to produce the mosaic in figure 3b out of the 30^{th} first images of the flower garden sequence. **We used the following parameter values:** $\alpha = 0.5$**,** $\beta = 20$ **and** $\lambda = 1$.Original images contain a tree in foreground that occludes parts of the background. The mosaicking reconstructed a view that completely discards it, and that indicates by black regions the background areas for which no information is available. Concerning sharpness, fig. 3b can be compared with the mosaic obtained by a masked median combination (fig. 3a). The difference is more visible on zoomed parts fig. 3 d and c : the white bar and the flower-bed appear much more neatly on the mosaic produced using the partitionning method.

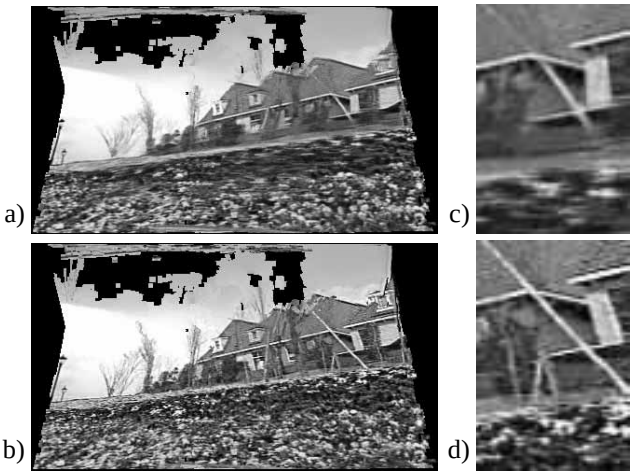

Figure 3. Mosaics obtained using the exposed framework (see text for explanations).

5. Conclusion

In this report a framework was proposed to produce a background mosaic from a video sequence. Our method involves two steps: 1) alignment of images and localisation of foreground objects, 2) pasting of the images onto the mosaic.

The structure of the first step relies on the assumption that the background is the dominant object in the source image , so that a dominant approach can be used (see section 2). Localisation of foreground objects is based on local residual motion intensity between aligned images. These two modules are run in an iterative refining process where alignment is computed on the segmented region, and segmentation use aligned images. The use of an a-priori background mask derived from a previously aligned image pair still improved the stability of the framework.

Concerning the mosaicking step (section 3), we reviewed two classical methods that do not take into account the presence of moving objects, and extended them to eliminate foreground objects. A new framework was proposed to achieve sharper mosaics, using a striping depending on the distance to foreground objects.

The whole framework was implemented and tested on the well-known flower-garden sequence. The proposed mosaicking method revealed itself as achieving a sharper mosaic than combining methods. What limits the whole framework is the algorithm we chose for object detection and that relies on the dominant motion assumption.

Applications of such an algorithm can be found in video indexation, where a whole sequence can be summed up in a single mosaic image that represents its background.

References

[1] J. R. Bergen and P. Anandan. Hierarchical computationnaly efficient motion estimation algorithm. In *Journal of the Optical Society of America A.*, 1987.

[2] J. Davis. Mosaics of scenes with moving objects. In *Proceedings of CVPR*, June 1998.

[3] M. Irani, P. Anandan, and S. Hsu. Mosaic based representation of video sequences and their applications. In *Fifth International Conference on Computer Vision*, pages 605–611, June 1995.

[4] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, 1992.

[5] R. Mégret and C. Saraceno. Building the background mosaic of an image sequence. Technical Report PRIP-TR-060, PRIP, TU Wien, 1999.

[6] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatio-temporal segmentation based on region merging. *IEEE-PAMI*, September 1998.

[7] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *CVPR*, pages 338–343, June 1997.

[8] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha. Universal mosaicing using pipe projection. In *International Conference on Computer Vision*, 1998.

[9] H. S. Sawhneg, S. Hsu, and R.Kumar. Robust video mosaicing through topology inference and local to global alignment. In *European Conference on Computer Vision*, pages 103–119, 1998.

[10] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing Special Issue: Sequence Compression*, 3(5):625–638, September 1994.