DYNAMIC VIDEO MOSAICING AND AUGMENTED REALITY FOR SUBSEA INSPECTION AND MONITORING

E. Trucco, A. Doull, F. Odone, A. Fusiello, D. Lane
Ocean Systems Laboratory
Department of Computing and Electrical Engineering
Heriot-Watt University, Edinburgh, EH14 4AS
Scotland, UK

Abstract

This paper reports a powerful technique for building panoramic mosaics from video sequences automatically. No information about the camera motion nor on its optical parameters are necessary. Mosaics can be built even in the presence of objects moving in front of the target scene (dynamic mosaicing), which are deleted by motion analysis. The technique also makes *augmented reality* possible, that is, inserting new elements in a video sequence under the correct perspective. Some results are included and discussed, and a URL where to find further results, color mosaics and MPEG sequences given.

1 Introduction

1.1 What is a mosaic?

A *mosaic* is a panoramic image of a scene formed by several overlapping images, each capturing only a portion of the scene. Mosaics are a useful way to represent the information of a video sequence: since sequence frames have often significant overlap, a mosaic of the sequence can provide significant compression rates. There are several possible descriptions of a scene that can be chosen depending on the scene in exam [1]:

Salient still [2, 3]. Static mosaics have been previously referred as *salient stills* of simply *mosaics*. They are usually built in batch mode, by aligning all frames of a sequence to a reference coordinate system. Static mosaics can be efficient representations for video storage and retrieval. The same techniques used for mosaicing can be also used for image stabilization, video compression, and content-based layered representation of information [4].

Dynamic mosaic. A limitation of static mosaics is that they are in constructed in batch mode: mosaic construction cannot begin before all frames are loaded. An alternative is to build *dynamic mosaics*, the contents of which is variable and constantly updated in time with the information of the current frame. When the first frame is read, the mosaic will coincide with the frame itself. In the further steps, the mosaic will be updated in order to be coherent with the latest frame read [1, 5].

Multiresolution mosaic. Changes in image resolution can occur within a sequence if the camera zooms significant in or out. A mosaic built at low resolution contains less information than the original sequence; on the other hand, a the mosaic at the highest resolution in the sequence would oversample low-resolution frames. This problem can be handled with a *multi-resolution* structure with captures information from each new frame at its highest resolution level.

1.2 What can mosaics be used for?

Video mosaicing has recently attracted a growing interest from the subsea robotics community, but also in the fields of automatic indexing of video data [6] , video coding, video editing, and virtual reality [7].

In the subsea domain, mosaics of sidescan sonar images are well known. Their construction is relatively simple thank to strong assumptions on the motion of the sensor. Video mosaics of subsea sequences have several applications in marine biology and geophysics, defence, surveying [8] mapping and autonomous navigation[9].

1.3 About this paper

This paper is structured as follows. Section 2 introduces mosaic building, homography estimation, and some applications. Section 3 sketches the feature tracking algorithm we adopted. Section 4 details mosaic building with a single motion, and Section 5 does the same for the case of multiple motions. Section 7 shows some results. Section 6 sketches how to build augmented reality mosaics. A short summary closes the paper.

2 Technical background

2.1 How is a mosaic built?

In this paper, we adopt the following algorithm. (1) We track the motion of special points, or *features*, across the sequence. (2) We use the position of corresponding features in different frames to work out the image transformation, or warping, aligning two frames correctly. We assume that this transformation is a *homography* [10], that is, a simple matrix operator producing the coordinates of a feature in a frame from its coordinates in a different one. Notice that this can be regarded as computing the *motion* of image points through the sequence, assuming that a matrix multiplication is a valid motion model.

Various other methods exist; algorithmic differences impact tracking, motion analysis, and frame alignment algorithms. [11, 7] discuss the pros and cons of various methods.

2.2 Homography estimation

Consider an image sequence with negligible parallax, which means that subsequent frames are approximately related by a homography. Assume that a set of corresponding points (features) have been tracked through the sequence (Section 3).

Four points, no three of them collinear, determine a unique homography. Indeed, eight independent parameters are required to define the homography. Two corresponding points (u, v), (u', v') in frames I and I' respectively provides two equations:

$$\begin{cases} u'(H_{3,1}u+H_{3,2}v+H_{3,3}) = H_{1,1}u+H_{1,2}v+H_{1,3} \\ v'(H_{3,1}u+H_{3,2}v+H_{3,3}) = H_{2,1}u+H_{2,2}v+H_{2,3} \end{cases} . \tag{1}$$

It is then necessary to find at least four point correspondences to define the transformation matrix up to a scale factor. Equation (1) can be rearranged as:

$$\begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -uu' & -vu' & -u \\ 0 & 0 & 0 & u & v & 1 & -uv' & -vv' & -v' \end{bmatrix} \begin{bmatrix} H_{1,1} \\ H_{1,2} \\ H_{1,3} \\ H_{2,1} \\ H_{2,2} \\ H_{2,3} \\ H_{3,1} \\ H_{3,2} \\ H_{3,3} \end{bmatrix} = \mathbf{0}. \tag{2}$$

For $n \geq 4$ points, we obtain a rank-deficient system of homogeneous linear equations, which has the form $\mathbf{Lh} = \mathbf{0}$. If $n > 4$ there are more equations than unknown, and, in general, only a least-squares solution can be found.

3 Tracking

This section describes the tracker we adopted [12]. Consider an image sequence $I(\mathbf{m}, t)$, with $\mathbf{m} = [u, v]^\top$, the coordinates of an image point. If the time sampling frequency is sufficiently high, we can assume that small image regions are displaced but their intensities remain unchanged:

$$I(\mathbf{x}, t) = I(\delta(\mathbf{m}), t + \tau), \tag{3}$$

where $\delta(\cdot)$ is the *motion field*, specifying the *warping* that is applied to image points. The fast-sampling hypothesis allows us to approximate the motion with a translation, that is, $\delta(\mathbf{m}) = \mathbf{m} + \mathbf{d}$, where $\mathbf{d}$ is a displacement vector. The tracker's task is to compute $\mathbf{d}$ for a number of selected points for each pair of successive frames in the sequence.

As the image motion model is not perfect, and because of image noise, Eq. (3) is not satisfied exactly. The problem is then finding the displacement $\hat{\mathbf{d}}$ which minimizes the SSD residual:

$$\epsilon = \sum_{\mathcal{W}} \left[I(\mathbf{m} + \mathbf{d}, t + \tau) - I(\mathbf{m}, t)\right]^2 \tag{4}$$

where $\mathcal{W}$ is a small image window centered on the point for which $\mathbf{d}$ is computed. By plugging the first-order Taylor expansion of $I(\mathbf{m} + \mathbf{d}, t + \tau)$ into (4), and imposing that the derivatives with respect to $\mathbf{d}$ are zero, we obtain the linear system $\mathbf{G}\mathbf{d} = \mathbf{e}$, where

$$\mathbf{G} = \sum_{\mathcal{W}} \nabla I \, \nabla I^{\top}, \quad \mathbf{e} = -\tau \sum_{\mathcal{W}} I_t \nabla I, \tag{5}$$

with $= \nabla I = [\partial I / \partial u \; \partial I / \partial v]^{\top}$ and $I_t = \partial I / \partial t$. Using this linear approximation of the solution, the Newton-Raphson iterative algorithm for minimizing (4) writes:

$$\begin{cases} \mathbf{d}_0 = \mathbf{0} \\ \mathbf{d}_{k+1} = \mathbf{d}_k + \hat{\mathbf{d}}. \end{cases}$$

where $\mathbf{d}_k$ is the displacement estimate at iteration k and $\hat{\mathbf{d}}$ is the solution of

$$\mathbf{G}\hat{\mathbf{d}} = \sum_{W} \Big[(I(\mathbf{m}, t) - I(\mathbf{m}{+}\mathbf{d}_k, t{+}1)) \, \nabla I(\mathbf{m}, t) \Big].$$

In this framework, a feature can be tracked reliably if a numerically stable solution to Eq. (3) can be found, which requires that $\mathbf{G}$ is well-conditioned and its entries are well above the noise level. In practice, since the larger eigenvalue is bounded by the maximum allowable pixel value, the requirement is that the smaller eigenvalue is sufficiently large. Calling λ_1 and λ_2 the eigenvalues of $\mathbf{G}$, we accept the corresponding feature if $\min(\lambda_1, \lambda_2) > \lambda$, where λ is a user-defined threshold [13].

The tracker produces a list of features coordinates for each image. For each couple of images, after all the features lost by the tracker have been discarded, the homography can be produced, using the method described in Section 2.2.

4 Mosaicing with single motion

The construction of a mosaic is accomplished in three stages: *motion estimation*, *registration* and *rendering*. Motion estimation has been described in Sections 2.2.

4.1 Frame registration

Once we have calculated the homography between image pairs, we must choose a common reference frame onto which to warp all image of the sequence. This can be done in two ways, depending on the amount of frame-to-frame overlap.

Frame to fixed frame registration. If the images do not change too much, that is, if the overlapping between an arbitrary pair of images is significant, a frame can be chosen as reference. The homographies to compute align each image to the reference frame.

Adjacent frames registration. If changes across sequence frames are significant, tracking is best done between contiguous frames. Transformation between non-contiguous frames, necessary to produce the global alignment, can be obtained by multiplying the transformation matrices of the in-between image frames.

4.2 Sequence alignment

In this case the $2D$ motion estimation and alignment of the image frames of the sequence can be performed in three ways [1].

Adjacent frames. The homographies are computed between successive frames of the sequence. They can be composed to obtain the alignment between *any* two frames of the sequence.

Frame to mosaic. To limit the problem of misalignments, for every new frame a temporary mosaic can be built and the new homography is computed between it and the new frame. This approach is alternative to the one of global alignment and further blending.

Mosaic to frame. If one wants to maintain each image in its coordinate system it can be better to align the mosaic to the current frame.

If a parallax-based $3D$ model is needed, given a sequence of images with full correspondences between adjacent ones, one can compute, for each view, the plane homography and the relative affine structure, using the previous view as the "second view", and then warp it to a reference view (the "third" view) using the appropriate view parameters, that is the new correspondences between the current image and the reference one.

4.3 Mosaic rendering

Once the images have been aligned, they can be integrated (or *blended*) into a mosaic using a temporal filter. Such filter produces the intensity of a mosaic pixel from the intensities of all corresponding pixels in the sequence. Possible filters include the following (see [1] for a review).

The *temporal average* of the intensity values. Moving objects would leave a "ghost-like" trace into the mosaic. This is effective for removing temporal noise.

The *most recent information*, that is, the entire content of the most recent frame is used to update the mosaic.

The *temporal median* of the intensity values. Here, moving objects with intensity patterns stationary for less than half of the sequence tend to disappear. In practice, moving objects are treated as outliers. The results are sharper than the ones obtained with temporal average.

5 Mosaicing with multiple motions

This section describes a method to segment moving objects, in order to build a mosaic of the background only.

After constructing the mosaic with feature-based registration, moving objects are segmented out by computing the grey-level differences between the stable background (mosaic) and the current frame. Similar approaches to ours have been used in the field of surveillance and targeting, where the egomotion of the camera is compensated before extracting moving targets from the background. For instance, in [14, 5] the motion is computed for every pixel with a robust technique, and outliers masks give the moving object. In [15] temporal analysis of gray levels, based on probabilistic models and a-priori information, is carried out in order to segment moving objects.

The motion of the background, that is, the relative motion of the camera with respect to the scene, can be estimated with a robust technique [16]. The idea is to identify pixels belonging to moving objects as outliers of the main motion field, that is, the homography of the background. This assumes that moving objects are not too big. Then, sequence registration can be performed in the usual way.

To segment out moving objects a suitable temporal filter must be chosen, for instance the median or the weighted median. The blending stage mosaics the whole background, while the moving objects disappear. A synthetic sequence of the background without the moving object can be obtained with a mosaic-to-frame registration (that is, a back registration of the mosaic onto every single frame of the image sequence)

A difference-based technique has been found effective for our purposes. Grey-level differences are computed between each original frame and the equivalent virtual one. The result is thresholded to obtain a binary map. The binary motion map identifies the image regions correspinding to moving objects, plus smaller blobs due to misalignments, change in illumination or noise.

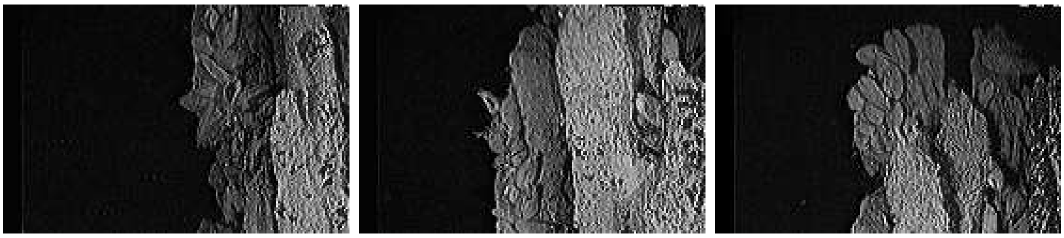

Figure 1: First, central and last frame from a sequence of a benthic structure (courtesy of IFREMER).

To segment out only the objects in motion, we assume for simplicity that only one object was moving in the scene. This is not strictly necessary, as long as moving pixels are (significantly) fewer than background ones. We detected the object in the first frame by choosing the area of the binary map containing the bigger connected component of moving pixels. After this initialisation, for every frame i, the centroid of the largest connected component of its binary map is computed. The connected component of the $(i+1)-th$ binary map chosen is the closest to the previous centroid.

At this point post-processing of the resulting maps is also needed, in order to obtain good quality segmentations. The morphological operator *closure* [17], that is *dilation* and *erosion* in cascade, produce a more compact blob, without adding noise and without altering its original dimension.

6 Augmented reality

Here, we use *augmented reality* to indicate *content-based editing of a video sequence*, typically adding a synthetic object, such as a banner (see next section for an example), to a background mosaic with or without moving object re-instated. The idea is to edit the background mosaic to add the object, then use a decoding procedure to create a new realistic sequence. The insertion of the synthetic object is done on a *metrically rectified* mosaic, that is, after warping the mosaic onto a convenient plane which makes the object addition simple and geometrically consistent [4]. After editing, the rectified mosaic is then warped back onto its original plane, and the synthetic objec appears automatically in the correct perspective.

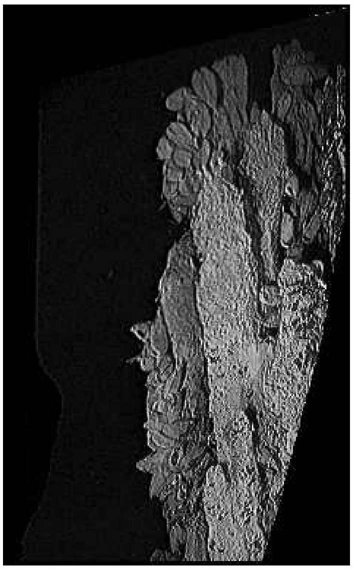

Figure 2: Mosaic of the sequence in the previous figure.

7 Results

Figure 1 shows frames 0, 70 and 138 from a sequence acquired by VICTOR, a ROV developed and operated by IFREMER, during a dive in the Pacific. This sequence contains a single relative motion between camera and scene. No information about camera nor vehicle motion was available. Figure 2 shows the mosaic of the sequence. Notice the warping of the individual frames, compensating for zoom and camera rotation.

Figure 3 is another result with a single-motion sequence (not shown), this time acquired by a hand-held commercial camcorder. Again no information on motion or camera parameters was known.

Figure 4 shows frame 0, 20 and 40 of a sequence containing two different motions. A commercial camcorder was moved by hand to track a car. The motion of the car pixels is different from that of the background pixels. Figure 5 shows the mosaic of the background from the whole sequence obtained after removing the car from the sequence by motion segmentation. Figure 6 shows an example of augmented reality:

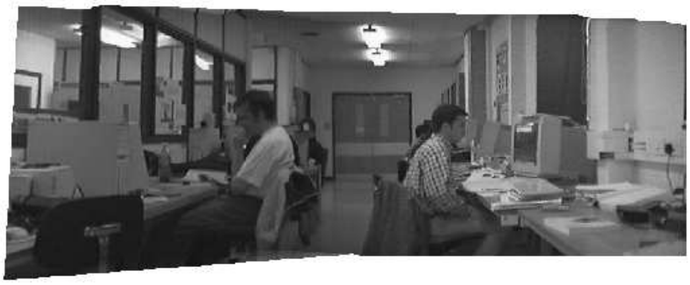

Figure 3: Mosaic of a laboratory sequence.

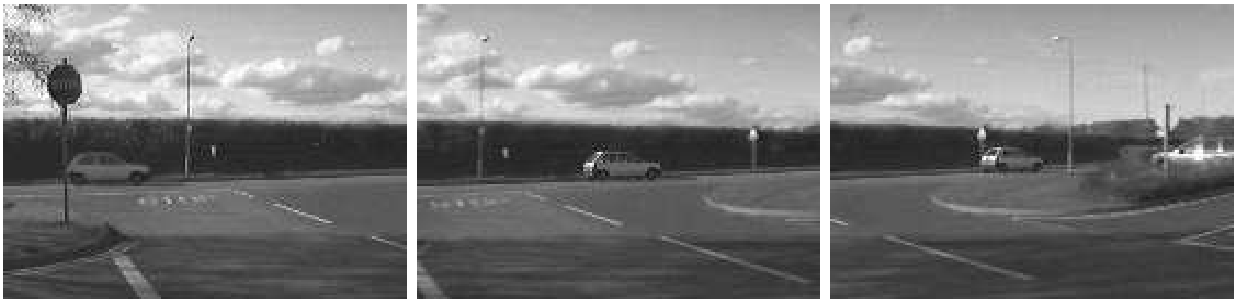

Figure 4: First, middle and last frames of a sequence containing two different motions.

a Heriot-Watt University banner has been inserted in the sequence (here, only one frame is shown). The correct perspective is computed automatically.

These and other examples, including colour mosaics and MPEG sequences, can be found at http://www.cee.hw.ac.uk/~fusiello/mosaic_demo.

8 Conclusions

We have presented a powerful technique for building panoramic mosaics from video sequences automatically, without information about the camera motion or its optical parameters. Mosaics can be built from sequences containing one or several relative motion between camera and scene. Robust motion analysis and frame differencing are used to remove moving objects and build background mosaics, onto which the

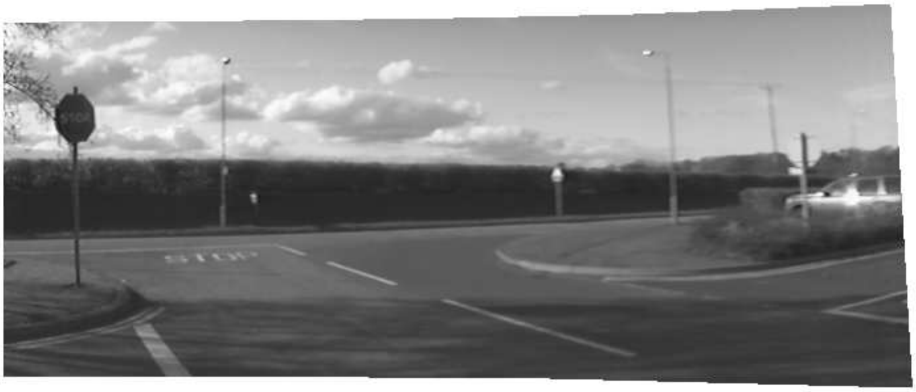

Figure 5: Mosaic of the car sequence, after remobing the car by motion analysis.

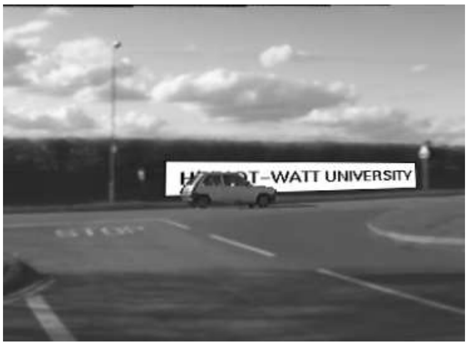

Figure 6: A sample frame of the "augmented reality" sequence.

moving objects can then be re-instated, showing them in motion across the panoramic background image. Using the same algorithms, s ynthetic elements can also be added to a video sequence and appear in the correct perspective.

Acknolewdgements

This research was partially supported by a EU Marie-Curie Fellowship (FAIR-GT97-3072), and by an EPSRC Visiting Fellowship (GR/M40844). We are indebted to Fabrice Lecornu of IFREMER for supplying the VICTOR sequences.

References

[1] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S Hsu. Efficient representations of video sequences and their applications. *Signal processing: Image Communication*, 8(4), 1996.

[2] M. Massey and W. Bender. Salient stills: process and practice. *IBM Systems Journal*, 35(3,4), 1996.

[3] S. Mann and R. W. Picard. Virtual bellows: Contructing high quality stills from video. In *IEEE International Conference on Image Processing*, 1994.

[4] F. Odone and A. Fusiello. Applications of 2d image registration. Technical Report RM-99-15, Dept Of Computing and Electrical Engineering, Heriot-Watt University, 1999.

[5] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.

[6] R. Brunelli, O. Mich, and C. M. Modena. A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, 10:78–112, 1999.

[7] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.

[8] Nuno Gracias and Joseé Santos-Victor. Automatic mosaics creation of the ocean floor. In *Proceedings of the OCEANS Conference*, 1998.

[9] S. Negahdaripour, X. Xu, and A. Khamene. Applications of direct 3D motion estimation for underwater machine vision systems. In *OCEANS'98 IEEE Proceedings*, pages 51–55, Nice, France, September 1998.

[10] J. G. Semple and G. T. Kneebone. *Algebraic projective geometry.* Oxford University Press, 1952.

[11] Sing Bing Kang. A survey of image-based rendering techniques. Technical Report CRL 97/4, Digital Equipment Corp., 1997.

[12] T.Tommasini, A.Fusiello, V. Roberto, and E. Trucco. Robust feature tracking in underwater video sequences. In *Proceedings IEEE Oceans*, pages 46–50, Sept 1998.

[13] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[14] I. Cohen and G. Medioni. Detecting and tracking moving objects in video surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:319–325, 1999.

[15] P.R. Giaccone and G.A. Jones. Segmentation of global motion using temporal probabilistic classification. In *British Machine Vision Conference*, pages 619–628, 1998.

[16] P. J. Rousseeuw and A. M. Leroy. *Robust regression & outlier detection.* John Wiley & sons, 1987.

[17] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.