MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Using Space-Time Interest Points for Video Sequence Synchronization

Daniel Wedge, Du Huynh and Peter Kovesi
School of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, W.A. 6009, Australia.
Email: {dwedge,du,pk}@csse.uwa.edu.au

Abstract

We introduce an algorithm for synchronizing two video sequences recorded by stationary cameras. It extends common RANSAC-based approaches that recover either a homography or a fundamental matrix from putatively matched spatial features in two images. In our algorithm, we detect space-time interest points in each sequence which represent events such as objects changing direction, and putatively matching points from each sequence are determined. A nested RANSAC framework on these putative matches is then used to firstly recover the frame offset and ratio of frame rates of the two sequences, then either a homography or a fundamental matrix relating the two views, depending on the type of motion contained within the sequences. No camera calibration or object tracking is required. Real sequences containing motion either on a plane or in free space are synchronized and it is demonstrated that this approach is successful in recovering the ratio of frame rates, the frame offset, and the homography or fundamental matrix relating the two sequences.

1 Introduction

An increasing number of computer vision applications are being developed that process multiple videos recorded simultaneously from different locations. Some applications of multiple view video analysis include comparisons of human motion [12], virtualized reality [6] and reconstruction of non-rigid scenes [14]. Video synchronization is essential to ensure consistency in the structure recovered in these applications.

Synchronization involves finding the temporal relationship between two or more video sequences. Most literature focuses on a linear model, where there is a temporal offset Δ between the sequences, and the ratio of frame rates is denoted by α . This can be expressed mathematically by:

$$t' = \alpha t + \Delta, \tag{1}$$

where t and t' are frame numbers of frames from each sequence recorded at the same instant in time.

Synchronization can be performed in hardware, for example, by embedding a timestamp in the video stream or sending a synchronization signal to cameras [6] though this can be costly and must be set up prior to recording. Alternatively, software algorithms can recover synchronization from visual cues.

Video synchronization algorithms can be divided into two classes: direct alignment and feature-based alignment. Direct alignment algorithms use pixel intensity data from frames of each video sequence for synchronization, for example, by comparing pixel intensities between sequences [2,15]. Feature-based synchronization algorithms use detected features for synchronization, for example, frame-to-frame object motion, or object trajectories throughout an entire sequence.

Many feature-based synchronization methods employ multiple view geometry methods. Reid and Zisserman [13] synchronized two short videos by recovering the homography of the ground plane and finding the frame offset that minimized the reprojection errors of moving points on the ground. Caspi and Irani [2] synchronized two sequences recorded by cameras with fixed internal parameters filming at a known ratio of frame rates. They used a RANSAC-based approach [3] to recover the frame offset and either a homography or fundamental matrix from multiple trajectory correspondences, aiming to minimize geometric reprojection error. Carceroni et al. [1] synchronized N video sequences (for $N \geq 2$) recorded by weakly calibrated cameras by establishing sets of tentative synchronized frames via the epipolar constraint, from which they recovered α and Δ values for all pairs of the N video sequences.

A fundamental matrix based algorithm by Pooley et al. [11] synchronized two sequences captured by moving cameras. The synchronization parameters were estimated via the Hough transform on a reparameterized parameter space of α and Δ, then refined using a gradient descent method to minimize the reprojection error.

Some algorithms use an algebraic measure for synchronization rather than a geometric error. Wolf and Zomet [16] constructed a measurement matrix from unmatched point trajectories in two sequences recorded at the same frame rate by affine cameras, using singular values beyond the expected rank of this matrix as a measure of synchronization. Tresadern and Reid [14] used a similar approach to also recover the frame rate ratio of two sequences where multiple trajectory correspondences were known. Rao et al. [12] used singular values of a measurement matrix to synchronize two sequences recorded by weakly calibrated perspective cameras. They synchronized two sequences of the same action performed by different people at the same location by computing a frame-to-frame mapping between sequences.

Rather than using frame-to-frame motion or object trajectories for synchronization, Yan and Pollefeys [17] used space-time interest points [8] as features for synchronization. Space-time interest points are reviewed in Section 2. Their algorithm recovered the temporal offset of two sequences recorded at the same frame rate from the distribution of interest points in each sequence. At each integer frame offset, they correlated the histograms of the distribution of interest points; the offset yielding the highest correlation score was declared the actual frame offset.

Laptev et al. [9] employed space-time interest points for detecting periodic motion in a single sequence. They used a RANSAC-based approach to firstly propose a period length and then fit a *dynamic fundamental matrix* from two proposed periodically equivalent point pairs.

Our algorithm synchronizes two sequences recorded by stationary cameras with fixed intrinsic parameters. Space-time interest points are detected in each sequence, and putatively matching interest points are proposed. Then, a two-step nested RANSAC approach firstly proposes a *temporal model*, i.e., α and Δ, then recovers the best spatial model for the proposed temporal model. The *spatial model* is either a homography for sequences containing planar motion, or a fundamental matrix if free object motion occurs. In contrast to Yan and Pollefeys' space-time interest point-based algorithm [17], our algorithm matches points between sequences and recovers the spatial model relating the two sequences. Our algorithm is similar to the periodic motion detection algorithm by Laptev et al. [9] in that a temporal model is proposed and a spatial model is fitted. However, we synchronize two views of the same event recorded at different frame rates instead of searching for constant-rate periodic motion. We present results for synchronizing real video sequences and demonstrate that our approach is successful.

2 Space-time interest points

Space-time interest points are locations in video sequences where a large variation in pixel intensities exists in space (within each frame) and time (between frames). They may be considered to be the equivalent in video sequences of spatial interest points in still images, e.g., Harris corners [4]. Interest points are often detected where and when an object has a significant force applied to it or where objects appear to merge or separate. Occlusions and dis-occlusions may also generate interest points. In most cases, interest points detected due to these events in one sequence would be detected at a different time and location in the other sequence, or they may not be detected at all. However, it is expected that the following step to determine putative matches will not match these interest points generated by occlusions, and even if putative matches were generated, the later RANSAC step would classify these points as outliers because they would not be consistent with the temporal or spatial models recovered from a set of interest points arising from actual events.

To detect space-time interest points, a second moment matrix μ is constructed for each pixel location (x,y) in each frame t of a sequence S:

$$\mu = \begin{bmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{bmatrix},$$

where L_ξ is the first order derivative in the ξ dimension of the Gaussian smoothed video sequence, computed via:

$$L_\xi(x,y,t;\sigma^2,\tau^2) = \frac{\partial}{\partial \xi}(g(x,y,t;\sigma^2,\tau^2) * S),$$

and $g(x,y,t;\sigma^2,\tau^2)$ is a separable Gaussian kernel with independent spatial and temporal variances, denoted by σ^2 and τ^2 respectively, given by:

$$g(x,y,t;\sigma^2,\tau^2) = \frac{\exp(-(x^2+y^2)/(2\sigma^2)-t^2/(2\tau^2))}{\sqrt{(2\pi)^3\sigma^4\tau^2}}.$$

Then, interest points are located at positive local maxima of the corner function H:

$$H = \det(\mu) - k \operatorname{tr}^3(\mu),$$

where Laptev suggests using $k \approx 0.005$. Positive local maxima of H correspond to (x,y,t) locations where the three eigenvalues of μ are significant.

Full derivations of the detection of space-time interest points are given by Laptev [8].

3 Determining putative matches

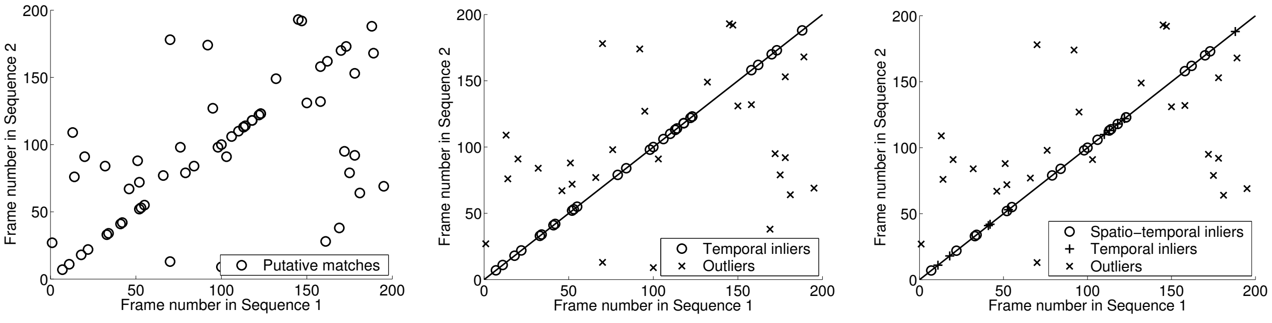
Putatively matching space-time interest point pairs are proposed by firstly computing a descriptor for each interest point, and then determining putative matches from the descriptor vectors. *Local jets* are vectors computed from the derivatives of image intensity gradient information around a point [7,10]. We constructed descriptor vectors from local jets computed over 3 spatial scales and 3 temporal scales. At each spatio-temporal scale, the local jet was computed from Gaussian derivatives up to order four, providing 34 derivative values. Hence, the multi-scale descriptor contained 306 elements. This approach follows that of Laptev and Lindeberg [10].

Laptev and Lindeberg proposed a number of distance measures for comparing these descriptors [10]; of these, we used the Euclidean distance between two descriptor vectors. A recursive winner-takes-all approach was used to determine putative matches, where at each instance, two points, $\mathbf{x}_i$ and $\mathbf{x'}_j$, were declared as a putative match if they had the greatest scalar product of any pair of points containing either $\mathbf{x}_i$ or $\mathbf{x'}_j$. All matches involving $\mathbf{x}_i$ and $\mathbf{x'}_j$ were then removed from further consideration, and the process repeated until no further matches remained.

4 Recovering the synchronization

RANSAC [3] is a random sampling method for fitting a model to a data set containing outliers. One example is recovering a homography from a set of putatively matching points [5], whilst determining which matches are inliers and which are outliers. Our synchronization algorithm employs two nested instances of RANSAC. The first, outer instance recovers the temporal model, whilst the inner instance estimates either a homography or a fundamental matrix, for sequences containing planar motion and free motion respectively. Our algorithm is summarized as follows:

1. Firstly, two pairs of putatively matching space-time interest points are randomly selected. Let t_i and t'_j denote the temporal components of two putatively matching space-time interest points. For each selected putative match, a *temporal component pair*, a 2D point (t_i, t'_j), is constructed. Fig. 1(a) shows an example of the distribution of temporal component pairs. A straight line is then fitted to the temporal component pairs of the two selected putative matches, yielding a proposed gradient α, and the



(a) (b) (c)

Figure 1: The process of determining spatio-temporal inliers from putative matches. (a) Firstly, temporal component pairs are constructed for all putative matches. (b) A straight line is fitted to two randomly selected temporal component pairs, and the temporal inliers determined. (c) The temporal inliers are used to fit a spatial model, from which spatio-temporal inliers, a subset of the temporal inliers, are determined.

y-intercept Δ, from Eq. (1).

2. Next, the inliers for the proposed temporal model are determined. For each temporal component pair, we compute the perpendicular distance to the previously computed straight line. Putative matches with temporal component pairs lying within a threshold distance from this line are *temporal inliers* as they are consistent with the values of α and Δ proposed in Step 1 (see Fig. 1(b)). The threshold is determined empirically. For each proposed temporal model, RANSAC requires the number of inliers to be counted to determine the best model. Our algorithm does not return the number of temporal inliers; rather, it uses the temporal inliers to initialize another instance of RANSAC to fit a spatial model.
3. We use RANSAC to fit a spatial model to the spatial components of the temporal inliers. To fit a homography, 4 temporally inlying putative matches are randomly selected and a homography estimated from the spatial components of these points. Alternatively, a sample of 8 temporal inliers can be selected so as to recover a fundamental matrix via the linear method. Further details are provided by Hartley and Zisserman [5]. A second distance threshold, independent of the threshold used in Step 2, is used to determine inliers from reprojection errors, again determined empirically. The RANSAC instance in this step may propose many spatial models for the given temporal model; the spatial model with the greatest number of inliers is returned.
4. The fourth step is to determine the number of inliers for the proposed temporal model and best corresponding spatial model. As only temporal inliers are used to fit the spatial model, the inliers to the recovered spatial model are a subset of the temporal inliers. Let the inliers to the spatial model be known as *spatio-temporal inliers*. In Fig. 1(c), it is shown that not all temporal inliers are spatio-temporal inliers. We use the number of spatio-temporal inliers as the number of inliers for the temporal model proposed in Step 1.

This process is repeated until the RANSAC algorithm determines that sufficient iterations have been completed. The temporal model and corresponding spatial model with the most spatio-temporal inliers are then returned.

5 Results

The algorithm was tested on pairs of sequences recorded by stationary cameras with fixed internal parameters. In the `shadow` series of sequences, the shadow of a moving object was projected onto a textured planar surface, and a homography relating the two views was recovered along with the temporal model. The `park` sequences contained free motion in 3D space, hence a fundamental matrix was recovered in place of a homography. Each video was recorded at 15 or 30 frames per second and contained between 125 and 540 frames; the resolution of each frame was 200 × 150 pixels. The 200 most significant space-time interest points were detected in each sequence. In detecting these points, the video sequence was convolved with a Gaussian with independent spatial and temporal variances. For all sequences, we used the temporal variance $\tau^2 = 2$; for the `shadow` sequences, we set the spatial variance to $\sigma^2 = 2$, whereas for the `park` sequences, we used $\sigma^2 = 4$.

The synchronization results shown in Table 1 confirm that this algorithm provides results comparable to manually synchronizing video sequences. Manual synchronization results are based on events such as a ball bouncing; hence, the frame offset can only be determined to ±0.5 frame. In Fig. 2, a frame from Sequence 1 is rectified such that it appears to have been viewed from the same location as Sequence 2. This visual comparison demonstrates that the recovered homography is accurate.

Fig. 3 shows synchronized frames from pairs of sequences containing free motion. The epipolar geometry has been recovered from the space-time features whose spatial components are illustrated in each view; epipolar lines corresponding to these points are overlaid in the other frame from each sequence. In one view of each sequence pair, the other camera is visible, and the recovered epipole is located close to the imaged location of the camera, indicating that the recovered epipolar geometry is satisfactory.

The results show that the algorithm presented here is successful in accurately recovering the temporal model and either a homography induced by a plane or a fundamental matrix relating the two views. We expect that the localization of the epipoles as shown in Figs. 3(c) and (d)

Table 1: The results show that the recovered frame rate ratio, $\hat{\alpha}$, and frame offset, $\hat{\Delta}$, are comparable to the manually recovered frame rate ratio and frame offset, denoted by $\overline{\alpha}$ and $\overline{\Delta}$ respectively. The `shadow` sequences contain motion on a plane, and the `park` sequences contain free motion.

Sequence pair	$\overline{\alpha}$	$\hat{\alpha}$	$\overline{\Delta}$	$\hat{\Delta}$
`shadow6`	1	1.0060	0.0	0.0513
`shadow7`	1	0.9993	0.0	–0.0543
`shadow8`	1	0.9983	0.0	0.0146
`shadow9`	1	1.0010	0.0	0.5776
`shadow10`	0.5	0.4958	94.0	94.9566
`park1`	1	0.9999	–10.0	–9.8060
`park5`	1	1.0019	0.0	–0.3589
`park9`	1	1.0001	24.5	23.7990
`park12`	1	0.9989	105.5	105.8763
`park15`	1	1.0006	–29.5	–29.2951

would be improved if the features used to recover the fundamental matrix were more evenly distributed spatially.

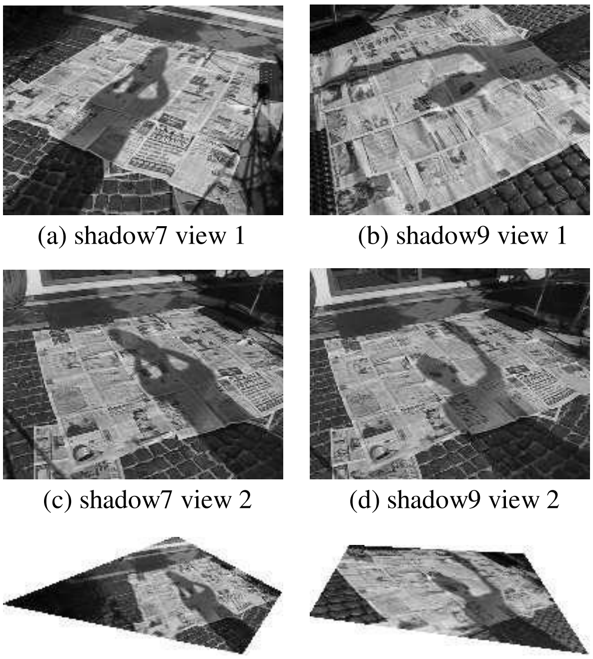
The time taken to synchronize the `shadow6` to `shadow8` sequences using a MATLAB implementation of this algorithm averaged between 4 and 5 minutes per pair of sequences on a 3GHz Pentium IV with 1GB of RAM. These sequences contained between 125 and 150 frames. On average, the detection of the space-time interest points consumed approximately 94% of the computation time.

6 Discussion

In the process of estimating the spatial and temporal models, the temporal model is proposed first, and then the spatial model is recovered from the set of temporal inliers. In sequences containing a significant number of outliers, this ordering is important. As the probability of selecting an outlier in a sample set increases with the size of the sample set, it is desirable to choose a smaller sample set. In this algorithm, this is achieved by firstly fitting a temporal model that can be proposed from only two putative matches, rather than a spatial model which requires at least four matches.

A further point of interest is how the inner model is affected when the sample used to propose the outer model contains an outlier. It is expected that if a temporal model is proposed from a pair of putative matches where at least one match is an outlier, there will not be many temporal inliers. In fact, there may be insufficient temporal inliers from which to propose a spatial model, in which case the temporal model is immediately discarded. If we were to firstly attempt to recover the spatial model, then there are a number of possible temporal models that could be proposed from the spatially inlying points (as it is assumed that all points used to propose the spatial model are *spatial inliers*). Hence, it is likely that the algorithm would propose many temporal models from the spatial inliers proposed from an incorrect spatial model, which is clearly inefficient.

It is noted that the accuracy of the estimated spatial model is heavily dependent on the estimated temporal model at each RANSAC iteration. Whilst an inaccurate temporal model may produce many temporal inliers, these

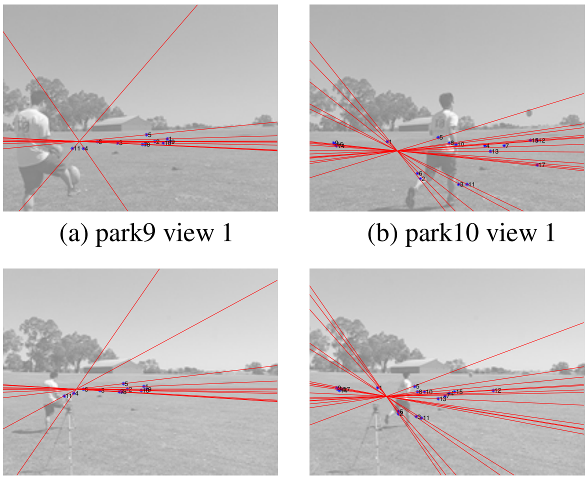


(a) shadow7 view 1 (b) shadow9 view 1

(c) shadow7 view 2 (d) shadow9 view 2

(e) shadow7 rectified view 1 (f) shadow9 rectified view 1

Figure 2: The result of synchronizing sequences containing planar motion. The pair of images (a) and (c) were recorded at the same instant in time, as were the images (b) and (d). The rectified views shown in (e) and (f) are the result of applying the recovered homography to the images in (a) and (b) such that those images appear to have been viewed from the same viewpoints as (c) and (d) respectively.



(a) park9 view 1 (b) park10 view 1

(c) park10 view 2 (d) park10 view 2

Figure 3: The result of synchronizing sequences containing free object motion. In each image, the spatial locations of inlying space-time interest points are displayed, and epipolar lines are overlaid corresponding to the spatial locations of inlying space-time interest points in the other sequence. In (c) and (d), the camera that captured frames (a) and (b) can be seen atop a tripod; the recovered epipole is close to the actual camera location. The pair of images (a) and (c) were recorded at the same instant in time, as were the images (b) and (d).

temporal inliers are unlikely to yield a consistent spatial model, hence the number of spatio-temporal inliers is expected to be low. This is not a problem, as due to the nature of RANSAC, it is expected that there are sufficient correct putative matches such that the correct temporal and spatial models will be recovered in at least one iteration.

In some applications, e.g., video surveillance, the cameras may be mounted in a fixed position and the spatial model may already be known. Hence, it is clearly not necessary to recover the spatial model as described in Section 4. In this case, the algorithm could be modified to ensure that the putative matches satisfy the supplied spatial model. Then, the method described in Section 4 would be simplified such that only the temporal model is recovered via Steps 1 and 2.

Another synchronization algorithm based on space-time interest points was developed by Yan and Pollefeys [17]. Their algorithm synchronizes a pair of sequences with a known ratio of frame rates, by constructing a histogram of the number of interest points occurring in each frame of each sequence. Then, the synchronization is recovered by correlating the two histograms at each integer frame offset; the offset yielding the highest correlation score is returned as the recovered frame offset.

A potential drawback of their algorithm is that it does not attempt to find matching space-time interest points from each sequence. Rather, it assumes that an event viewed by multiple cameras will produce a similar number of space-time interest points in corresponding frames of each video sequence. However, view-dependent events such as occlusions and dis-occlusions may generate interest points at different time instants of the two video sequences, or interest points in one video sequence only. This may significantly affect the correlation score, and hence the recovered frame offset. Our algorithm is more robust in that interest points detected for such events are unlikely to be declared as putative matches, and further, because these events are unlikely to occur at the same time and place in both sequences, they will not satisfy the temporal and spatial models and will hence be discarded.

A possible case where our algorithm may fail includes sequences containing repeated or periodic motions, which often generate many similar space-time interest points in both sequences. Consequently, the number of outliers would be excessive and incorrect temporal and spatial models may be returned by the RANSAC fittings. This problem is known to be common to all video synchronization algorithms [1,12–17].

7 Conclusion

It has been shown that our proposed algorithm successfully synchronizes pairs of video sequences without requiring object tracking. The accuracy of the recovered frame offset and frame rate ratio are comparable with manual synchronization, and it has been confirmed visually that the recovered homography or fundamental matrix relating the two sequences is accurate.

Acknowledgements

Daniel Wedge wishes to acknowledge the support of an Australian Postgraduate Award. The authors wish to thank the anonymous reviewers for their constructive comments.

References

[1] R. L. Carceroni, F. L. C. Pádua, G. A. M. R. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1:746–753, 2004.

[2] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, Nov 2002.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, Jun 1981.

[4] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[6] I. Kitahara, H. Saito, S. Akimichi, T. Onno, Y. Ohta, and T. Kanade. Large-scale virtualized reality. In *CVPR Technical Sketches*, 2001.

[7] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987.

[8] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[9] I. Laptev, S. J. Belongie, P. Perez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *Proceedings of the International Conference on Computer Vision*, 2005.

[10] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *ECCV Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[11] D. W. Pooley, M. J. Brooks, A. J. van den Hengel, and W. Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. In *Proceedings of the International Conference on Image Processing*, 2003.

[12] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *Proceedings of the International Conference on Computer Vision*, 2003.

[13] I. D. Reid and A. Zisserman. Goal-directed video metrology. In *Proceedings of the European Conference on Computer Vision*, pages 647–658, 1996.

[14] P. A. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. In *Proceedings of the British Machine Vision Conference*, 2003.

[15] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *Proceedings of the European Conference on Computer Vision*, pages 538–550, 2006.

[16] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision*, 68(1):43–52, 2006.

[17] J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In *Proceedings of Advanced Concepts for Intelligent Vision Systems*, 2004.