Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Video synchronization from human motion using rank constraints

Philip A. Tresadern [a,*], Ian D. Reid [b]

[a] University of Manchester, School of Cancer and Imaging Sciences, Imaging Science and Biomedical Engineering, Stopford Building, Oxford Road, Manchester, Greater Manchester M13 9PT, United Kingdom
[b] Active Vision Lab, Robotics Research Group, University of Oxford, Parks Rd, Oxford OX1 3PJ, United Kingdom

ARTICLE INFO

Article history:
Received 30 April 2007
Accepted 8 March 2009
Available online 25 March 2009

Keywords:
Video synchronization
Structure from motion

ABSTRACT

This paper presents a method of synchronizing video sequences that exploits the non-rigidity of sets of 3D point features (e.g., anatomical joint locations) within the scene. The theory is developed for homography, perspective and affine projection models within a unified rank constraint framework that is computationally cheap. An efficient method is then presented that recovers potential frame correspondences, estimates possible synchronization parameters via the Hough transform and refines these parameters using non-linear optimization methods in order to recover synchronization to *sub-frame accuracy*, even for *sequences of unknown and different frame rates*. The method is evaluated quantitatively using synthetic data and demonstrated qualitatively on several real sequences.

1. Introduction

In order to recover non-rigid human motion, commercial systems (e.g., [1]) employ a number of hardware-synchronized and accurately calibrated cameras under controlled studio conditions. High contrast markers at anatomical locations on the surface of the body are tracked in each camera, their 3D coordinates computed by triangulation and a ‘skeleton’ fitted to the resulting marker set using kinematic constraints. Various motion parameters (e.g., joint angles) can then be estimated over the sequence.

A more practical system would eliminate many of these constraints such that human motion can be recovered from stock footage using only a few cameras that are unsynchronized and uncalibrated. This would not only reduce the cost and technical complexity of the solution but also extend its application to fields such as sporting analysis. For example, a single sequence of a sporting event may include an action observed from several viewpoints, possibly at different speeds (e.g., a slow motion action replay). In such cases, neither camera synchronization nor calibration is available and a method that recovers these parameters from the data itself is required.

For this to be achieved, however, the complete system must address four key problems: recovering projected anatomical landmarks; establishing spatial correspondence; camera synchronization (the focus of this paper); camera calibration.

The research literature on human tracking is vast (see [2] for a recent survey). Although some success has been achieved using model-based, multi-view tracking [3–5], the recovery of anatomical parameters (e.g., joint centres or angles) from a *monocular* sequence has only been partially addressed. Research methods showing some promise include database searching [6–8], regression [9–11], assembling kinematic structure from independently detected body parts [12–14], model-based tracking of the limbs [15–19] and tracking joints directly based on appearance models [20].

However, *tracking* the human body is not our goal in this paper. Therefore, in order to demonstrate the synchronization method independently of the tracking method employed, we use hand-labelled joint locations (e.g., ‘left shoulder’) that provide all the information required for feature localization and matching (i.e., spatial correspondence). Obviously, using hand-labelled feature trajectories is not feasible for *live* sequences, although a real-time 2D full body tracker would solve this problem. However, the very nature of live sequences implies that they are approximately synchronized anyway. We stress that *the presented synchronization method is applicable regardless of how feature locations are obtained*, although its accuracy is dependent on the quality of the input feature locations.

Camera synchronization (the focus of this paper) ensures that image features matched between *sequences* also correspond to the same instant in *time* before triangulation. In commercial systems, this is achieved using hardware although this adds to the cost and complexity of the solution. More importantly, most *stock* footage is captured *without hardware synchronization* such that an alternative solution is necessary for 3D reconstruction. However, several works have shown that the image data itself can provide sufficient constraints to synchronize the cameras [21–23]. In particular, our previous studies have shown this to be the case for sequences of human motion [24,25]. The ability to synchronize sequences from the available feature locations makes the method

[*] Corresponding author.
E-mail address: philip.tresadern@manchester.ac.uk (P.A. Tresadern).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.03.012

applicable to stock footage where hardware synchronization is unavailable.

Finally, camera calibration provides 3D structure in a Euclidean coordinate frame such that meaningful motion parameters (e.g., joint angles and body segment lengths) may be recovered, as demonstrated for two views in [26,27].

1.1. Video synchronization

Commercial motion capture systems use hardware to synchronize cameras by using a trigger signal from a single source such that every camera opens its shutter at the same time. This engineering solution to the problem increases the technical complexity and cost of the system. In contrast, we present an algorithm that aligns the sequences in time using the data itself, as illustrated by Fig. 1 for two unsynchronized cameras. For every frame in one sequence, our aim is to recover the corresponding frame in the other. Note that since the cameras are unsynchronized, there may be no frame in the second view that corresponds exactly to the selected frame in the first.

We assume that the temporal relationship between sequences is linear such that for a given frame, f, in one sequence the corresponding frame in the other sequence, f', is specified by

$$f' = \alpha \cdot f + \Delta, \tag{1}$$

where α is the ratio of the frame rates and Δ is the offset of the 0th frame in the second sequence with respect to the first. In all cases we seek to recover Δ to sub-frame accuracy and in some cases we also seek to recover α (although in many cases it is known that $\alpha = 1$ in advance). In the case of non-rigid motion, we pose the synchronization problem as a search for consistent *structure* between the two sequences.

1.2. Related work

An early form of synchronization was proposed by Reid and Zisserman [21] who manually synchronized two sequences using the geometric distance of a point in one view from its epipolar line in the other as a measure of temporal correspondence between two sequences. The assumption of known epipolar geometry (often computed using static background points that are common to both views) required for such a metric has also been applied in other recent methods [28–30]. However, for wide baseline sequences, background features often have a very different appearance between views or are not even visible in both cameras, thus making spatial correspondence difficult.

Synchronization has also been demonstrated for sequence pairs related by a homography, such as when the scene is planar [31,32] or when the cameras have coincident centres of projection [33]. Later work [22] synchronized sequences from the trajectory of a single point imaged under perspective projection by iteratively estimating spatial and temporal alignment parameters but assumed that the cameras do not move relative to each other. Furthermore, it is unclear how sensitive the algorithm performance was with respect to the values used to initialize the optimization. An alternative approach minimizing the distance between back-projected lines was recently proposed by Tuytelaars and Van Gool [23] although it was not clearly demonstrated how this improved on point-based methods.

Although many existing synchronization methods (including this one) can easily be extended for more than two views (albeit at some computational expense – see Section 4.1), algorithms based on the trifocal tensor have also been proposed specifically for the three view case [34]. Other methods have recovered a non-linear temporal warping (e.g., via dynamic programming) between sequences that were not captured simultaneously for action recognition applications [35].

Methods that begin by establishing putative frame correspondences typically use robust line-fitting methods such as RanSaC [31,24,30] or the Hough transform [28,25] to reduce the effect of gross outliers when estimating synchronization parameters.

The presented method is inspired by the work of Wolf and Zomet [36] who used rank constraints [37] of a matrix of image measurements to define its 'energy' above an expected rank bound. This energy is minimized when structure is most consistent between sequences (i.e., when they are synchronized). Therefore, the offset between sequences was found by a linear search across all possible offsets. Their method was novel in that it did not require exact point correspondences between views (although it did make the weaker assumption that the points tracked in the

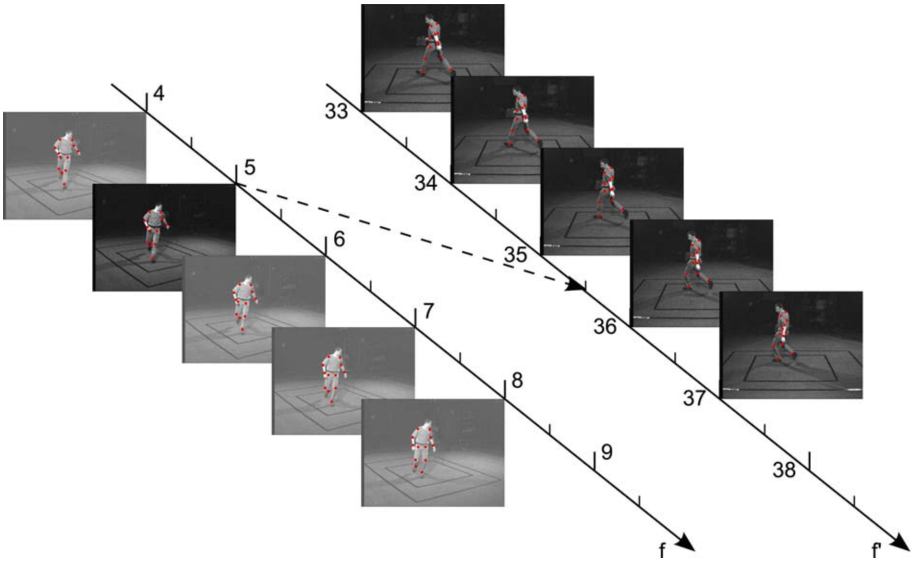

Fig. 1. Timelines of two sequences with synchronization offset indicated by the dashed arrow. Given a frame in one sequence (the darker frame), our goal is to find the corresponding frame in the other sequence. Note that, as shown in this example, an exactly corresponding frame may not exist due to a finite time interval between frame capture.

second sequence could be expressed as a fixed linear combination of a subset of points in the first sequence).

However, the limitation of their method was that it recovered synchronization by pooling data from groups of at least $3N+2$ frames, where N was the number of points tracked in the first sequence. As a result, their method required that the cameras had the same frame rate and were rigidly fixed with respect to each other. The pooling of data over multiple frames also fails when the scale of an object changes over time due to camera zoom or perspective effects. Furthermore, their rank bound that indicated temporal alignment was defined by a heuristic measure. Finally, they only recovered synchronization to the nearest frame.

In contrast, we develop a rank-based method to recover synchronization to *sub-frame* accuracy for sequences of *unknown and differing frame rates*. Structural consistency is evaluated *independently* for every frame pair, using a rank constraint that is related directly to an algebraic distance measure. In the affine case, this rank constraint is directly related to the geometric reprojection error and therefore has an intuitive interpretation.

Furthermore, evaluating each frame pair independently means that (i) the cameras are free to move independently of each other and (ii) the method can deal with temporary occlusion of feature points. Finally, our method does not require static background features to estimate the epipolar geometry of the cameras. This is an advantage in wide baseline applications where, in contrast to background features, foreground features (e.g., points local to the human subject) are more often within the view of both cameras and can be matched by geometric (rather than photometric) constraints.

1.3. Paper outline

This paper builds on our previous work [24,25] by presenting the theory and method in greater detail, including proofs where required. We also present a much more thorough evaluation of performance for both noiseless and noisy data for a quantitative and qualitative analysis, respectively.

We begin by developing a unified theory of rank-based synchronization for homography, perspective and affine projection models in Section 2. In particular, we show how this framework leads naturally to the well-known 'Factorization Method' [37] in the affine case. This theory is then employed within a computationally efficient algorithm that estimates putative frame correspondences, performs robust line fitting and optimizes the recovered synchronization parameters using non-linear error minimization (Section 4). Section 5 presents results of a number of evaluations using synthetic data and the method is demonstrated on real examples in Section 6. Since affine structure is also recovered as part of the synchronization process, we include a brief note on calibrating to a Euclidean coordinate frame in Section 7 before concluding in Section 8.

2. Theory

The basic idea underpinning our approach is simply stated: if the motion being observed is non-rigid, a metric that measures the consistency of the scene between views will assign a low cost to frames that are temporally aligned and a high cost to those that are not. This assumption of non-rigidity is common to most synchronization algorithms [21,31,36,23,28]. Algorithms that do not require non-rigidity instead impose other constraints, e.g., that the cameras must be rigidly attached to each other [32] or that close initial values of the synchronization parameters are given [22].

We investigate a non-rigidity metric for pairs of frames related by a homography, the fundamental matrix and the affine fundamental matrix. For further details on multiple view geometry, we direct the reader to [38].

2.1. Homography model

The case of recovering synchronization for sequences related by a homography was notably studied by Caspi and Irani using optical flow methods [32,33]. In contrast, we consider the case where two cameras observe *point* features (e.g., Harris corners) moving independently in a plane. Under this model, corresponding *homogeneous* image features, $\mathbf{z}$ and $\mathbf{z}'$, are related by a homography, $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \tag{2}$$

such that

$$\mathbf{Hz} = \mathbf{z}' \tag{3}$$

$$\Rightarrow \mathbf{z}' \times \mathbf{Hz} = \mathbf{z}' \times \mathbf{z}' = \mathbf{0} \tag{4}$$

up to scale (since the image points are homogeneous). Each point correspondence imposes two linear constraints on the homography such that, under ideal conditions, we can write the system of constraints for *all* points as

$$\mathbf{M}_H\mathbf{h} = \mathbf{0}, \tag{5}$$

where $\mathbf{M}_H$ is a $2N \times 9$ matrix of constraints defined by the image feature locations and $\mathbf{h} = (h_1, \ldots, h_8, 1)^T$. We then define the sum of squared *algebraic* distances, $d_{alg}(\cdot,\cdot)$, between features $\mathbf{z}'_i$ measured in a frame from sequence 2 and those transferred, $\mathbf{Hz}_i$, from a frame in sequence 1 as

$$\sum_i d_{alg}(\mathbf{z}'_i, \mathbf{Hz}_i)^2 = \|\mathbf{M}_H\mathbf{h}\|^2 \tag{6}$$

such that linear least squares methods can be employed to minimize d_{alg} for a given pair of frames. For $N \leqslant 4$ points, any $\mathbf{h}$ in the right nullspace of $\mathbf{M}_H$ satisfies (5) exactly. For $N > 4$ points, however, d_{alg} is minimized by setting $\mathbf{h} = \hat{\mathbf{h}}$, the right singular vector corresponding to the smallest singular value, σ_{min}, of $\mathbf{M}_H$ and rescaling appropriately. For this particular value of $\mathbf{H} = \hat{\mathbf{H}}$, it can be shown (see Appendix A) that

$$\sum_i d_{alg}\left(\mathbf{z}'_i, \hat{\mathbf{H}}\mathbf{z}_i\right)^2 = \sigma^2_{min}. \tag{7}$$

This suggests that a 'rank constraint' framework may be employed to synchronize sequences since a small value of σ^2_{min} indicates a potentially close temporal alignment between a pair of frames.

2.2. Perspective model

For perspective projection, we again propose using the *algebraic* distance measure in a rank-constraint framework as a computationally cheap alternative to minimizing a geometric distance (e.g., as in [22]). Corresponding homogeneous image features, $\mathbf{z}$ and $\mathbf{z}'$, are related by the perspective fundamental matrix, $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & 1 \end{bmatrix}, \tag{8}$$

where

$$\mathbf{z}^T\mathbf{Fz}' = 0. \tag{9}$$

In this case, however, each point correspondence imposes only a single linear constraint on $\mathbf{F}$ such that

$$\mathbf{M}_F\mathbf{f} = \mathbf{0}, \tag{10}$$

where $\mathbf{M}_F$ is a $N \times 9$ matrix of constraints defined by the image feature locations and $\mathbf{f} = (f_1, \ldots, f_8, 1)^T$. We define the sum of squared *algebraic* distances, $d_{alg}(\cdot,\cdot)$, between features $\mathbf{z}'_i$ from a frame in sequence 2 and their epipolar lines, $\mathbf{F}\mathbf{z}_i$, as computed from the corresponding features in sequence 1 as

$$\sum_i d_{alg}(\mathbf{z}'_i, \mathbf{F}\mathbf{z}_i)^2 = \|\mathbf{M}_F\mathbf{f}\|^2. \tag{11}$$

Again, linear least squares methods can be employed to minimize d_{alg} for a given pair of frames. For $N \leqslant 8$ points, any $\mathbf{f}$ in the right nullspace of $\mathbf{M}_F$ satisfies (10) exactly and a unique solution is provided by the well-known 'eight point algorithm' [39,40]. For $N > 8$ points, d_{alg} is minimized by setting $\mathbf{f} = \hat{\mathbf{f}}$, the right singular vector corresponding to the smallest singular value, σ_{min}, of $\mathbf{M}_F$ and rescaling appropriately. As with the homography case, for this particular value of $\mathbf{F} = \hat{\mathbf{F}}$ it can be shown (see Appendix A) that

$$\sum_i d_{alg}\left(\mathbf{z}'_i, \hat{\mathbf{F}}\mathbf{z}_i\right)^2 = \sigma^2_{min}, \tag{12}$$

again suggesting that a rank constraint framework may be applicable albeit at a cost of requiring twice as many points as the homography model.

2.3. Affine model

For most of this paper, however, we will focus on the simpler case of affine projection, a commonly used projection model in human motion analysis applications since the human body has limited depth and perspective effects are typically small at any given time. In the affine case, the fundamental matrix takes the form

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & a_1 \\ 0 & 0 & a_2 \\ a_3 & a_4 & 1 \end{bmatrix} \tag{13}$$

and again

$$\mathbf{M}_A\mathbf{a} = \mathbf{0}, \tag{14}$$

where $\mathbf{a} = (a_1, \ldots, a_4, 1)^T$. However, in this case the $N \times 5$ constraint matrix, $\mathbf{M}_A$, takes the particularly simple form

$$\mathbf{M}_A = \begin{bmatrix} x_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & x'_N & y'_N & 1 \end{bmatrix}, \tag{15}$$

where $(x_n, y_n)^T$ and $(x'_n, y'_n)^T$ denote the nth feature in the first and second view, respectively. As in the other projection models, linear least squares are employed such that $N = 4$ provides an exact solution whereas for $N > 4$ points, setting $\mathbf{a}$ equal to the right singular vector corresponding to σ_{min} minimizes the algebraic distance between the point sets.

2.4. Factorization approach

Furthermore, it can be shown (see Appendix B) that translating all points so that their centroid lies at the origin gives a new matrix, $\widetilde{\mathbf{M}}_A$ with a tighter lower bound on $rank(\widetilde{\mathbf{M}}_A)$. Under these conditions

$$\widetilde{\mathbf{M}}_A\tilde{\mathbf{a}} = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} & x'_1 - \bar{x}' & y'_1 - \bar{y}' \\ \vdots & \vdots & \vdots & \vdots \\ x_N - \bar{x} & y_N - \bar{y} & x'_N - \bar{x}' & y'_N - \bar{y}' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \mathbf{0} \tag{16}$$

such that $rank(\widetilde{\mathbf{M}}_A) \leqslant 3$ under ideal conditions.

In proposing the Factorization method [37], Tomasi and Kanade arrived at the same conclusion by different reasoning. For two affine views, their observation shows that the normalized $4 \times N$ 'measurement matrix' of image coordinates, $\mathbf{W}$, can be written as a product

$$\mathbf{W} = \begin{bmatrix} x_1 - \bar{x} & \cdots & x_N - \bar{x} \\ y_1 - \bar{y} & \cdots & y_N - \bar{y} \\ x'_1 - \bar{x}' & \cdots & x'_N - \bar{x}' \\ y'_1 - \bar{y}' & \cdots & y'_N - \bar{y}' \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} [\mathbf{Z}_1 \quad \cdots \quad \mathbf{Z}_N], \tag{17}$$

where $\mathbf{P}_i$ is the 2×3 projection matrix of the ith view and $\mathbf{Z}_n$ is the 3×1 vector of inhomogeneous 3D coordinates of the nth feature. Specifically, (17) shows that the rank of $\mathbf{W}$ is bounded above by 3 since it is a product of a 4×3 projection matrix (denoted $\mathbf{P}$) and $3 \times N$ structure matrix (denoted $\mathbf{Z}$). Note that for the two view case $\mathbf{W} = \widetilde{\mathbf{M}}_A^T$, thus confirming the rank constraints derived earlier.

However, in contrast to using the affine fundamental matrix, the factorization method naturally extends to any number of views. Tomasi and Kanade exploited this fact to propose the factorization of $\mathbf{W}$ into affine motion and structure using the Singular Value Decomposition (SVD), thus recovering all $\mathbf{P}_i$ and $\mathbf{Z}_n$ up to an affine transformation. Reid and Murray [41] later demonstrated that the Factorization method recovers the 'optimal' structure and motion in terms of minimizing reprojection error and can therefore be interpreted as a Maximum Likelihood estimate, assuming isotropic Gaussian noise.

Unlike the homography and perspective projection cases, affine projection is linear. Therefore, $\mathbf{PZ}$ gives the reprojected feature locations directly, in contrast to the non-linear projections (e.g., homography and perspective projection) where a rescaling of the homogeneous coordinates (to correct for depth) must be applied to obtain reprojected image features. As a result, the sum of squared *geometric* reprojection error, E, following factorization is also linear and obtained as

$$E = \|\mathbf{W} - \mathbf{PZ}\|^2_F, \tag{18}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Once again, it can be shown (see Appendix C) that this is directly related to the singular values of $\mathbf{W}$ by

$$E = \sum_{i=4}^{r} \sigma^2_i, \tag{19}$$

where we have assumed the singular values are in order of magnitude. In the two view case (where $r = 4$) this reduces to $E = \sigma^2_{min} = d_{alg}$. This equivalence between algebraic and geometric error (previously noted in [38]) makes a rank-constraint framework especially applicable under affine projection.

2.5. Affine vs. perspective projection

In the context of synchronization from human joint locations, we are faced with a choice of whether to use an affine or perspective projection model since both are equally valid. Most applications employ an affine model for simplicity since the relief of the human body is relatively small when compared to the typical distance of the subject from the camera. As a result, the affine approximation is a sufficiently accurate approximation in most cases.

Furthermore, the affine model has only four parameters that must be estimated in contrast to 8 for perspective projection. Therefore, for a fixed number of features the affine model is more strongly constrained. Alternatively, in cases where a features are unavailable (e.g., on an occluded limb) there may be enough remaining features to constrain the affine model but not the perspective model.

3. Rank-based synchronization

Intuitively this measure of non-rigidity would seem to be an appropriate metric for determining synchrony: when frames are temporally aligned, image correspondences are consistent with an underlying interpretation of three-dimensional structure (the pose of the person at that instant) and reprojection error is small. However, when the sequences are not aligned the images are of *different* points in space and therefore not subject to any rank constraint.

Using the results derived so far, we propose two cost functions in order to recover the synchronization between two sequences. The first match cost, $C_1(f,f')$, reflects the residual reprojection error resulting from the pairing of two frames, f and f':

$$C_1(f,f') = \sigma^2_{min}, \tag{20}$$

where σ_{min} is the smallest singular value of $\mathbf{M}_H$, $\mathbf{M}_F$ or $\mathbf{W}$. In the affine case, $\mathbf{W}(f,f')$ is defined as

$$\mathbf{W}(f,f') = \begin{bmatrix} \mathbf{W}(f) \\ \mathbf{W}(f') \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^f & \cdots & \mathbf{z}_N^f \\ \mathbf{z}_1^{f'} & \cdots & \mathbf{z}_N^{f'} \end{bmatrix} \tag{21}$$

and $\mathbf{z}_n^f$ and $\mathbf{z}_n^{f'}$ are the normalized image coordinates of the nth feature in frame f and f' of sequences 1 and 2, respectively. Pairs of frames with a low value of $C_1(f,f')$ are a good match whereas those with a high value of $C_1(f,f')$ are structurally inconsistent. This is apparent in Fig. 2a showing a plan view of the cost function, $C_1(f,f')$.

Having defined a match cost between frames from two different sequences, we then define a cost function for the synchronization parameters, α and Δ. The most intuitive is simply the sum of all errors over the entire sequence such that

$$C_2(\alpha,\Delta) = \sum_f C_1(f, \alpha \cdot f + \Delta). \tag{22}$$

This defines a cost surface (shown in Fig. 2b) upon which we find a local minimum via non-linear optimization based on a sensible initial estimate, as described in the following section.

4. Method

For every frame, f, in sequence 1 we compute the match cost for every potentially corresponding frame, f', in sequence 2. Fig. 3a shows $C_1(37,f')$, a 'slice' through the cost function at frame 37 of sequence 1. In this example, we see that multiple minima are present due to periodic motion in the action being performed (a running motion in this case).[1]

Exhaustively computing $C_1(f,f')$ for all pairings of f and f' generates a coarse 2D cost surface as shown in Fig. 2a. Although this requires $F \times F'$ evaluations of C_1 for sequences of F and F' frames, the method is relatively efficient due to the simple form of the cost function.

From C_1, we select putative frame correspondences (Fig. 4a) via thresholding and non-minimum suppression across f and f'. In previous work [24], we used random sample consensus (RanSaC [42]) to fit a line to these potential frame correspondences in a robust manner. In this work, however, we use the frame correspondences to cast votes in a Hough accumulator [43] from which it is straightforward to extract peaks (Fig. 4b) corresponding to potential synchronization parameters. If α is known in advance, we compute a 1D Hough array that indicates potential offsets only.

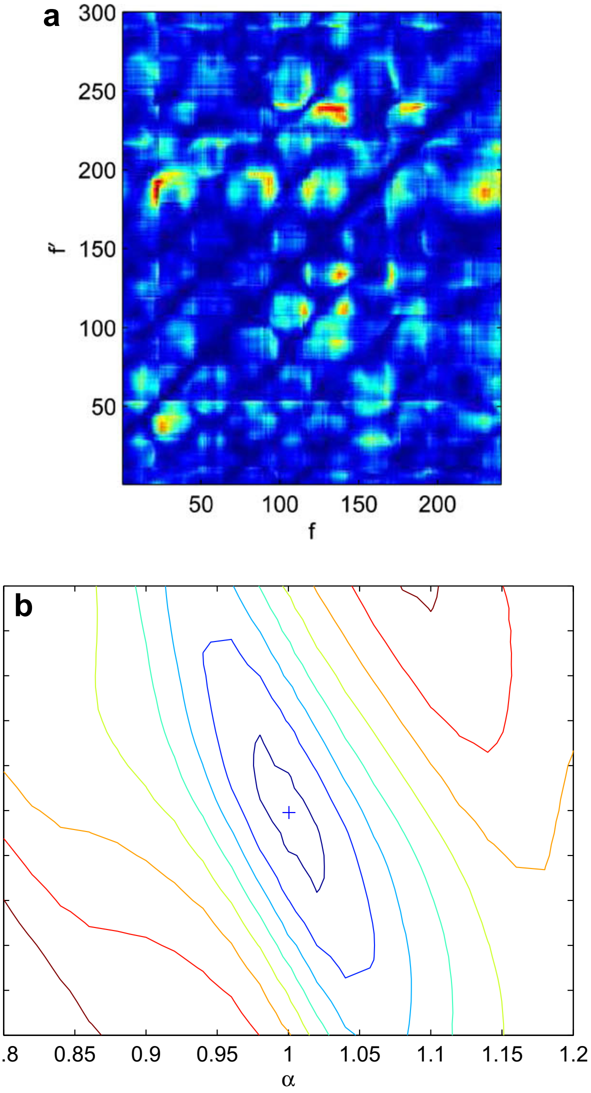

Fig. 2. (a) The cost surface $C_1(f,f')$ for the real running example, shown in plan view and normalized such that values range from 0 (dark) to 1 (light). Note the visible 'channel' close to the principal diagonal where the true correspondence lies. (b) Contour plot of $C_2(\alpha,\Delta)$, also indicating the solution recovered via non-linear optimization. From the elliptic shape of the basin of attraction, we see that errors in α may be compensated by a complementary error in Δ.

Since we expect there to be multiple peaks in ambiguous cases, we retain all peaks with a score greater than 80% of the maximum. Furthermore, we eliminate highly unlikely estimates that lie outside of the range $0.1 \leqslant \alpha \leqslant 10$.

Since the recovered correspondences are between whole frames, the Hough transform returns estimates of potential alignment whose resolution is limited by the bin size. Moreover, if α is known to be unity then the accuracy of Δ is theoretically limited to the nearest whole frame.

We can refine this initial estimate by optimizing the cost function $C_2(\alpha,\Delta)$ directly in order to recover α and Δ to *sub-frame* accuracy. This requires us to evaluate $C_1(f,f')$ using image features at frame f in the first sequence and from the corresponding (i.e., predicted) instant, f' in the second sequence. However, the predicted value, f' for an arbitrary α and Δ almost certainly has a real (i.e., non-integer) value. As a result, the required image features are not directly available since they occur *between* observed frames. Therefore, we approximate by interpolating between the two nearest observed frames:

[1] Pooling data from consecutive frames can resolve ambiguity, as demonstrated in [36,24]. However, since this imposes further constraints on the system (namely that the cameras have the equal frame rates and do not move relative to each other) we do not pursue this further in the current work.

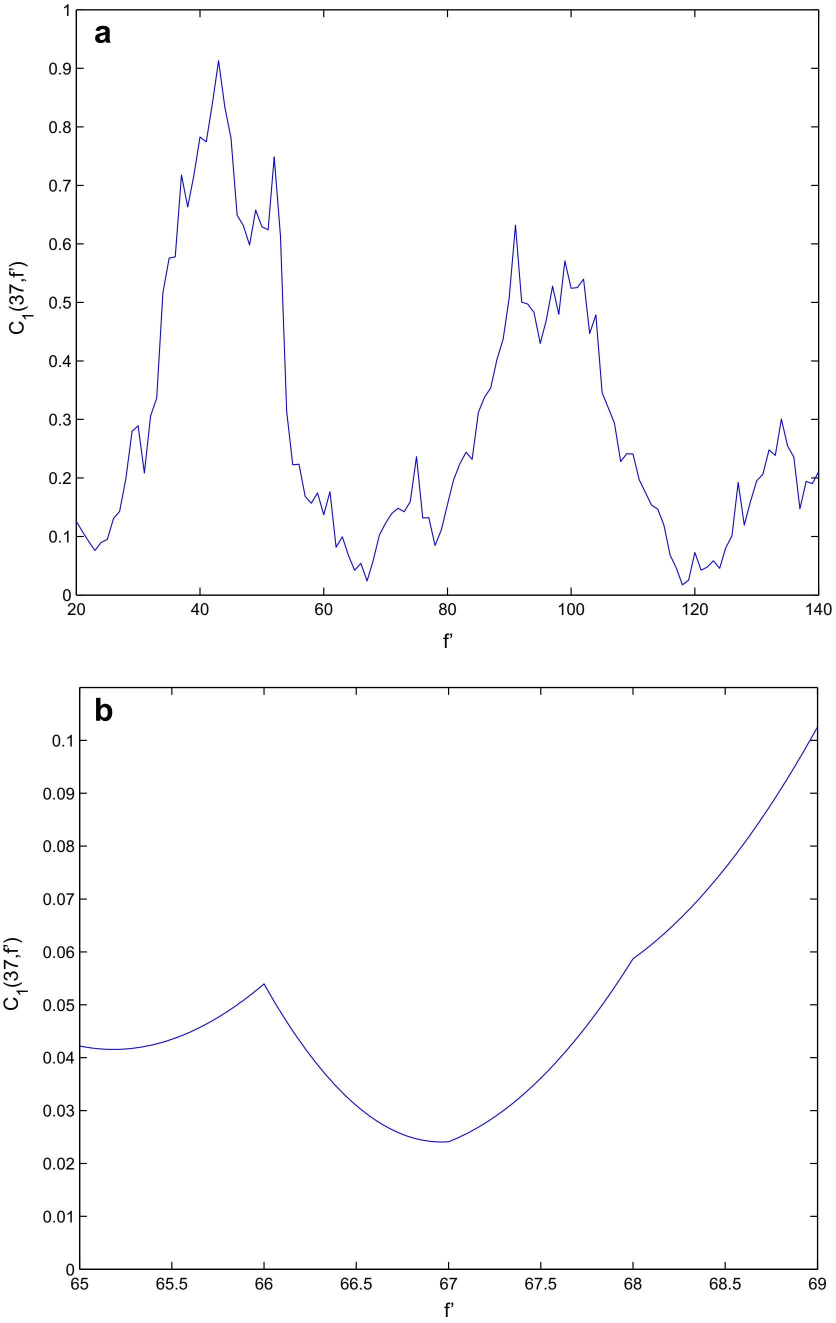

Fig. 3. (a) Plot of $C_1(37, f')$ for the running sequence. Note that in addition to the correct minimum (frame 67, in this case) another minimum is evident (frame 118) due to periodic motion (also noted by [36]). (b) $C_1(37, f')$ evaluated using interpolated feature locations in the interval [65,69]. The computed minimum is observed close to the correct minimum ($f' = 67$).

$$\mathbf{W}(f') = (1-\beta)\mathbf{W}(\lfloor f' \rfloor) + \beta\mathbf{W}(\lceil f' \rceil), \tag{23}$$

where

$$\beta = f' - \lfloor f' \rfloor \tag{24}$$

is a linear coefficient that weights the two nearest frames appropriately. Linear interpolation was found to reduce reprojection errors compared with single frame accuracy (see Section 5.4) although higher order interpolation (e.g., quadratic, cubic) may yield superior estimates.

Using this interpolation, we are able to evaluate $C_1(f, f')$ for non-integer values of f' (see Fig. 3b). In particular, we can evaluate $C_1(f, f')$ for *every* measured frame, f, and its predicted counterpart, f'. This then allows us to evaluate $C_2(\alpha, \Delta)$ for any (α, Δ) pair via Eq. (1).

We therefore evaluate $C_2(\alpha, \Delta)$ for each of the (α, Δ) pairs that were retained from the Hough accumulator after thresholding. We then refine the (α, Δ) pair with the smallest error using standard optimization methods (the Nelder-Mead Simplex algorithm, implemented as `fminsearch` in Matlab[2]). This recovers a locally

[2] Other optimization methods (e.g., Levenberg-Marquardt, Non-linear Conjugate Gradients) are equally applicable since the error surface is typically convex within the region of the correct solution. As long as the Hough transform returns a good initial value and the optimizer is well-tuned, the final result is the same regardless of the optimization method.

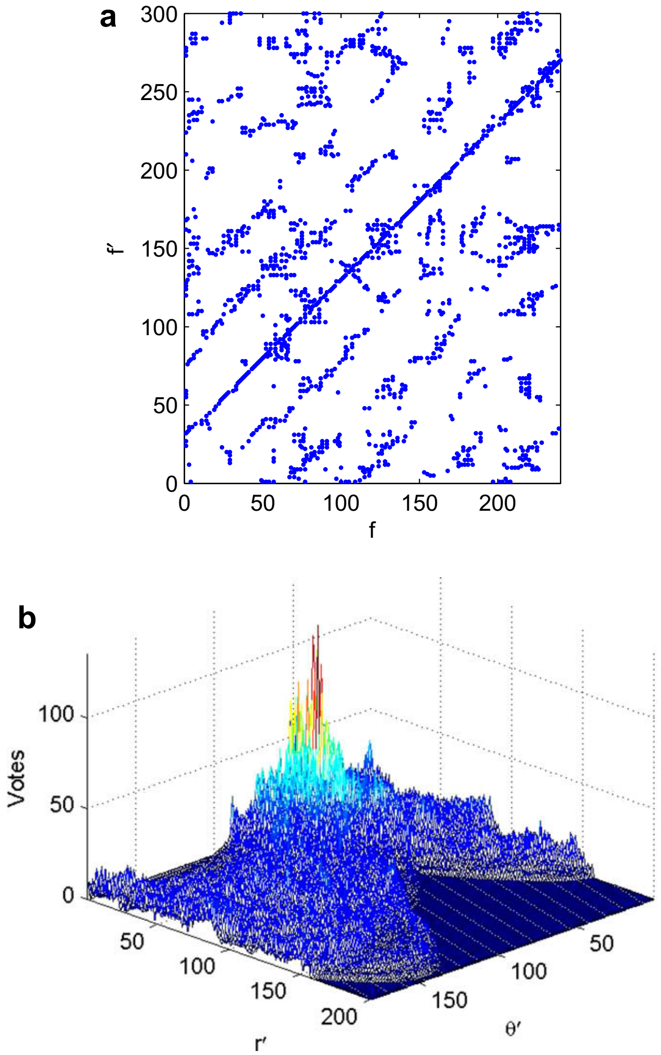

Fig. 4. (a) Local minima corresponding to potential frame correspondences, recovered using non-minimum suppression and thresholding of the cost surface shown in Fig. 2a. Note the high number of good matches along the diagonal where the true correspondence lies. (b) 3D surface of the accumulator array with a visible peak at $(\alpha, \Delta) = (1, 32.76)$.

optimal solution for the cost surface, $C_2(\alpha, \Delta)$, as illustrated in Fig. 2b. In all of our examples, $C_2(\alpha, \Delta)$ was convex within the region of the correct solution. However, we also show examples where the location of the minimum is not at its true value (Section 6.2) and where there are multiple minima (Section 6.3). In the latter case, it is essential that the Hough transform finds the best initial estimate.

This process can be seen as a hierarchical search for the globally optimal solution, using computationally cheap methods (the Hough transform) to reject a high number of poor estimates early on so that relatively expensive processes, such as computing the cost $C_2(\alpha, \Delta)$, are performed for only a small number of hypotheses.

4.1. Synchronizing multiple sequences

We digress for a moment to consider the synchronization of more than two sequences. In the general case of k sequences, we must search for a line in $\Re^k$ space. If each sequence has $\sim F$ frames, we must evaluate approximately F^k potential matches and the problem scales exponentially with the number of sequences. This is typically the case with any synchronization algorithm.

To make the search more computationally efficient, however, we propose an alternative approach. Using $k = 3$ as an example, we note that any line fit to correspondences between sequences 1 and 2 defines a plane in $\Re^3$. Similarly, any line fit to correspondences between sequences 2 and 3 also defines a plane in $\Re^3$. The intersection of these planes then completely defines the synchronization parameters and can be checked for consistency using the putative correspondences between sequences 3 and 1.

In general, the sets of synchronization parameters between k pairs of sequences can be searched for the combination that results in the most consistent solution. This approach would require approximately kF^2 frame comparisons and therefore becomes much more tractable. This solution can then be refined using non-linear optimization as before.

4.2. Outlier rejection

Since the method is based on a least-squares approach to determine the reprojection error for each putative frame pairing, gross outliers (e.g., as a result of tracking error) have a highly undesirable effect. Therefore, it is beneficial to make the system more robust by detecting such outliers and reject them at an early stage. This may be achieved using a strategy such as RanSaC to estimate the affine fundamental projection matrix from four matched points, then seek consensus from the remaining points. Solutions with low consensus as a result of gross outliers are rejected whilst those with high consensus are used to compute the reprojection error. Since this method has been successfully applied to estimate fundamental matrices for some time [44], we do not present results of its application in this work.

5. Performance evaluation

In the previous work [24,25], we evaluated the synchronization algorithm using real sequences made up of hand-labelled (and, therefore, noisy) features. In this evaluation, however, we employed a synthetic sequence pair of a human impersonating a monkey (Fig. 5). Since this sequence was synthetic and noiseless, we obtained a more accurate assessment of algorithm performance that was free from the effects of noisy input data. Performance on real sequences is presented in Section 6.

Two views, synchronized by design, each contained 480 frames of 14 point features located at anatomical landmarks on the body (shoulders, elbows, wrists, hips, knees, ankles, midriff and head) that were imaged under perspective projection. We then deleted 50 frames from the beginning and end of the first view to give ‘ground truth’ synchronization parameter values of $\Delta_{gt} = 50$ and $\alpha_{gt} = 1$. Except where stated, we fixed α at its known value and recovered only Δ, making the problem a one-dimensional search. Later experiments demonstrate the ability of the algorithm to recover both Δ and α. A quantitative analysis of sensitivity to noise was available by adding noise of a known variance; performance on real, noisy sequences is evaluated in Section 6. Comparisons with previously reported results are also included where appropriate.

5.1. Baseline performance

We begin with the simplest case where the offset took an integer value, α was constrained at unity and the image data were noiseless. However, we also note that an offset that is exactly integral typically occurs only for sequences captured with hardware-synchronized cameras; real sequences almost always have a real-valued Δ. However, it is useful for evaluation purposes to use synchronized sequences that have been offset by an integer value.

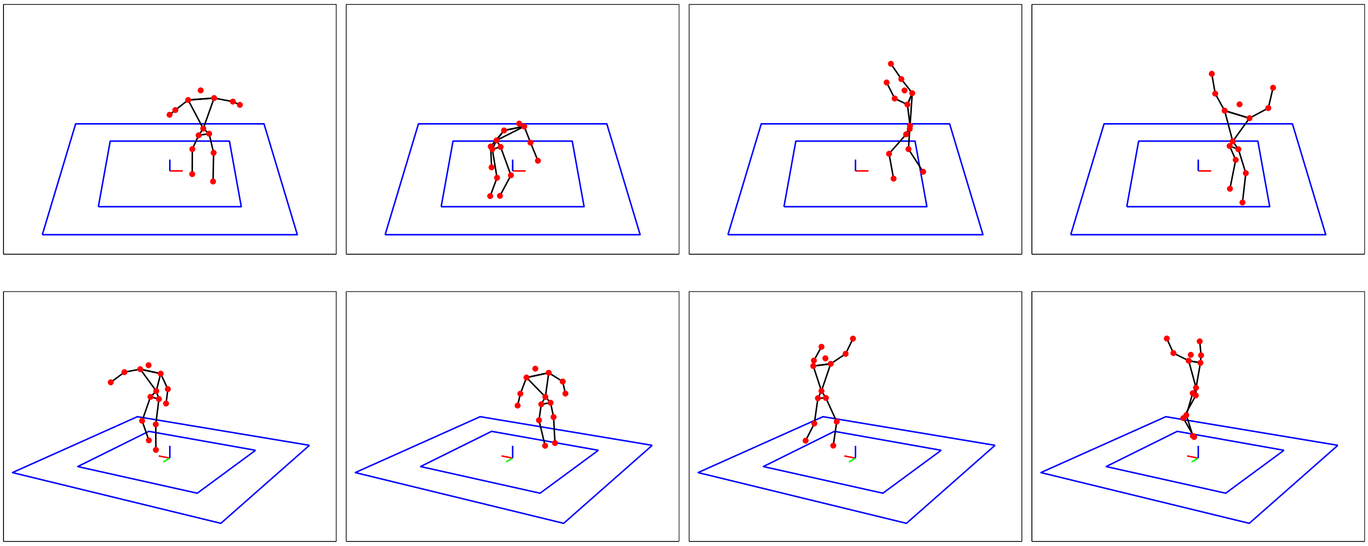

Fig. 5. Synthetic ‘monkey’ sequence as seen from two wide baseline viewpoints. The red circles indicate point features used as inputs to the synchronization algorithm.

When using the affine projection model, the initial offset recovered from the 1D Hough array was $\Delta = 50$. This estimate was then updated to $\Delta = 50.071$ by non-linear minimization of C_2. The error in Δ of 0.07 frames compensates for the error in reprojected feature locations as a result of using the incorrect projection model (i.e., affine rather than perspective).

For comparison, the perspective model (i.e., the true projection model) also recovered an initial offset of $\Delta = 50$ from the Hough accumulator. This was then unchanged by the non-linear minimization of C_2 since the data were noiseless and the correct projection model was used. Similar results were reported for a real sequence in previous work [25].

We also measured the time taken for each of the processing steps (Table 1). From these data, we see that evaluating the matching cost between frame pairs accounted for most of the processing time. Furthermore, the greater complexity of the perspective projection model is apparent in the processing time required both for matching and optimization.

5.2. Performance over varying temporal offset

To demonstrate the accuracy of the algorithm for sub-frame offsets, we synthesized unsynchronized sequence pairs with offsets of $\Delta_{gt} = 5, 5.1, \ldots, 5.9$ frames by taking interleaved frames from the available synchronized sequences. For example, we synthesized an offset of $\Delta_{gt} = 5.3$ frames by taking frames $1, 11, 21, \ldots$ from one sequence and frames $54, 64, 74, \ldots$ from the other.

Fig. 6 compares the recovered offsets with ground truth where it can be seen that the recovered offsets were typically accurate to within a few hundredths of a frame despite: (i) the assumption of linear motion between frames degrades for the low frame rates at which we are operating; (ii) lowering the effective frame rate reduces the number of frames available for estimation of the synchronization parameters. A similar accuracy was also observed for the real running sequence in previous work [24].

Table 1
Measured time for each stage of synchronization method.

	Time taken (s)	
	Affine	Perspective
Evaluate frame matching costs	9.01	23.45
Find potential matches	0.16	0.14
Compute Hough accumulator	0.29	0.57
Non-linear optimization (Δ only)	0.88	18.65
Non-linear optimization (Δ and α)	2.22	44.61

We note that this ability to recover accurate synchronization parameters from widely spaced frames suggests that a coarse-to-fine approach may be employed. In this example, the user could label only every 10th frame and the algorithm would find an initial solution to the synchronization. Putative frame correspondences, selected based on this initial solution, could then be labelled in order to refine the solution. This would provide substantial savings in time since only a fraction of the available data would require labelling for a satisfactory solution.

It is also interesting that the accuracy was high for offsets of $\Delta_{gt} = 5,\ \Delta_{gt} = 5.5$ and $\Delta_{gt} = 6$. In contrast, offsets of $5 < \Delta_{gt} < 5.5$ were underestimated and offsets of $5.5 < \Delta_{gt} < 6$ were overestimated. This is due to the fact that equally spaced points on a curve do not project to equally spaced points on the linear approximation. Instead, points are more closely spaced near the end-points of the linear approximation. This warping results in the exact effect that was observed. More complex interpolation models (e.g., quadratic polynomials, B-splines) may reduce this effect.

5.3. Sensitivity to noise

In the previous work [24], we investigated the effect of noise on recovering frame correspondences only. In this experiment, we pursue the matter further and investigate its effect on the final result following non-linear optimization.

The original image feature locations were perturbed by zero mean Gaussian noise of increasing standard deviation, σ_n pixels, for 50 tests at each level of noise. The scatter plot in Fig. 7a shows the recovered offsets as a function of the level of noise (the figure was typically ~ 300 pixels high in the image). Interestingly, we see that the distribution of the recovered offsets became multi-modal with increasing noise. Furthermore, the modes of the distribution typically occurred halfway between frames. We believed that this was due to the fact that for slow movements, where the inter-frame difference between corresponding features is small compared to the (Gaussian) noise, an interpolated feature location that averages halfway between two noisy estimates is closer to the true value than the measured position in *either* frame.

To investigate this hypothesis further, we interpolated features from the second sequence for a range of values of Δ using data that

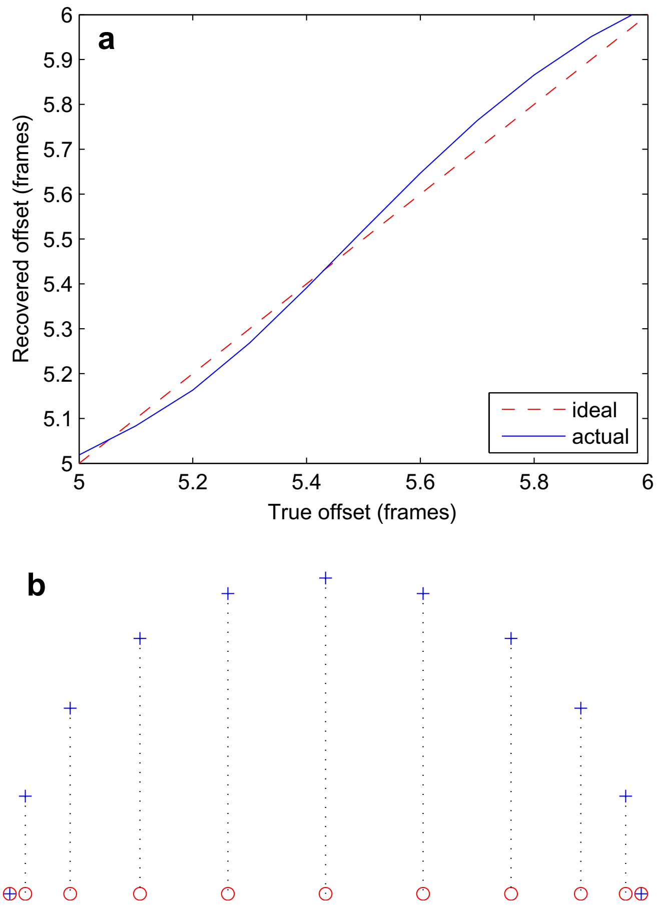

Fig. 6. (a) Recovered values for simulated offsets of $5, 5.1, \ldots, 5.9$ frames. We see the recovered offset is typically accurate to within hundredths of a frame. (b) Points equally spaced on a curve do not project to equally spaced points on the linear approximation. This warping results in underestimation or overestimation of Δ_{gt}, depending on its true value.

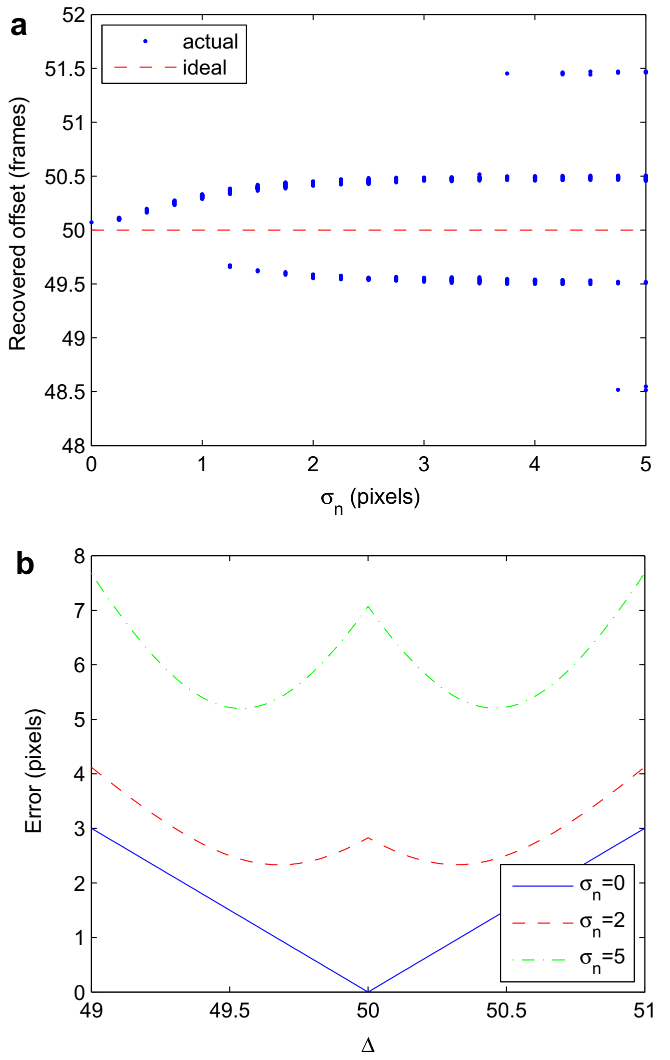

Fig. 7. (a) Recovered offsets over 50 trials at each level of added zero-mean Gaussian noise of standard deviation, σ_n. (b) Reprojection error with respect to Δ for data corrupted by noise of increasing variance.

was corrupted by noise of standard deviation $\sigma_n = 0$, $\sigma_n = 2$ and $\sigma_n = 5$. We then averaged the reprojection error between the interpolated features and their true values over 50 trials (Fig. 7b). For noiseless data, we see that reprojection error was minimized at the correct offset. However, for noisy data the reprojection was minimized by interpolating using a value of $\Delta = \Delta_{gt} \pm 0.5$, thus averaging between nearby frames and confirming our intuition.

5.4. Reprojection errors

In the baseline comparison, we demonstrated the ability of the algorithm to recover Δ to within a few hundredths of a frame when Δ_{gt} took an integer value. In this experiment, we evaluated the reprojection error having synchronized sequences with a *sub-frame* offset. In particular, we demonstrate that estimating the offset to sub-frame accuracy reduced the error between the interpolated feature locations and their true values. To quantify reprojection errors, we used odd frames from the first sequence and even frames from the second, giving parameter values of $\alpha_{gt} = 1$ and $\Delta_{gt} = 25.5$. From an initial estimate of $\Delta = 26$, a refined value of $\Delta = 25.53$ was recovered using the affine projection model. In comparison, using the perspective model resulted in a recovered offset of $\Delta = 25.501$.

For each frame, we then computed four sets of feature locations for the second view: features taken directly from the nearest frame of sub-sampled data (‘Nearest’); interpolated features using recovered synchronization parameters (‘Recovered’); interpolated features using known synchronization parameters (‘Known’); features taken directly from *original* image data (‘Original’). Since these feature locations are typically of full rank (i.e., not subject to the rank constraint) due to perspective, we also computed a reduced-rank version that satisfied the rank constraints by projecting onto the appropriate subspace. For each set of estimated features at every frame, $\mathbf{W}_{est}$, we then computed the sum of squared reprojection errors with respect to the original image data, $\mathbf{W}$:

$$E_{est} = \|\mathbf{W} - \mathbf{W}_{est}\|_F^2. \tag{25}$$

Table 2 shows the mean E_{est} over all frames, showing that sub-frame accuracy offers a considerable reduction in reprojection error com-

Table 2
Reprojection errors under the affine model where $r = rank(\mathbf{W}_{est})$. The results show a considerable reduction using sub-frame accurate alignment rather than the nearest frame. Results for the perspective projection model were similar.

	$r = 4$	$r = 3$
Nearest ($\Delta = 26.00$)	9.6204	9.6878
Recovered ($\Delta = 25.53$)	1.7728	2.9072
Known ($\Delta = 25.50$)	1.6334	2.8266
Original	0	2.2990

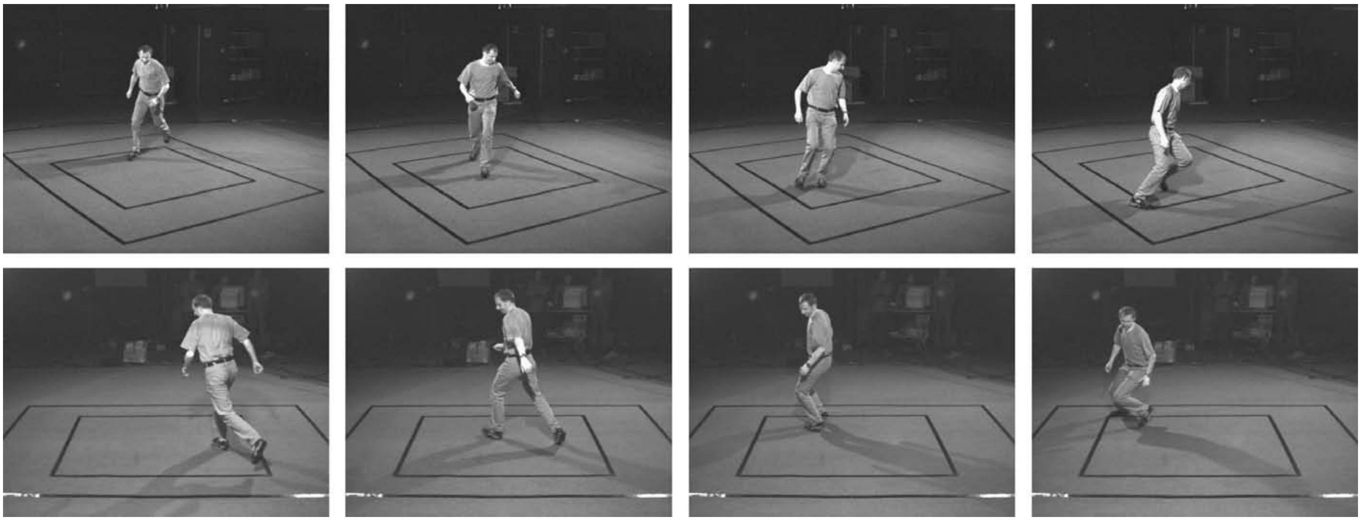

Fig. 8. Running sequence as seen from two wide baseline viewpoints.

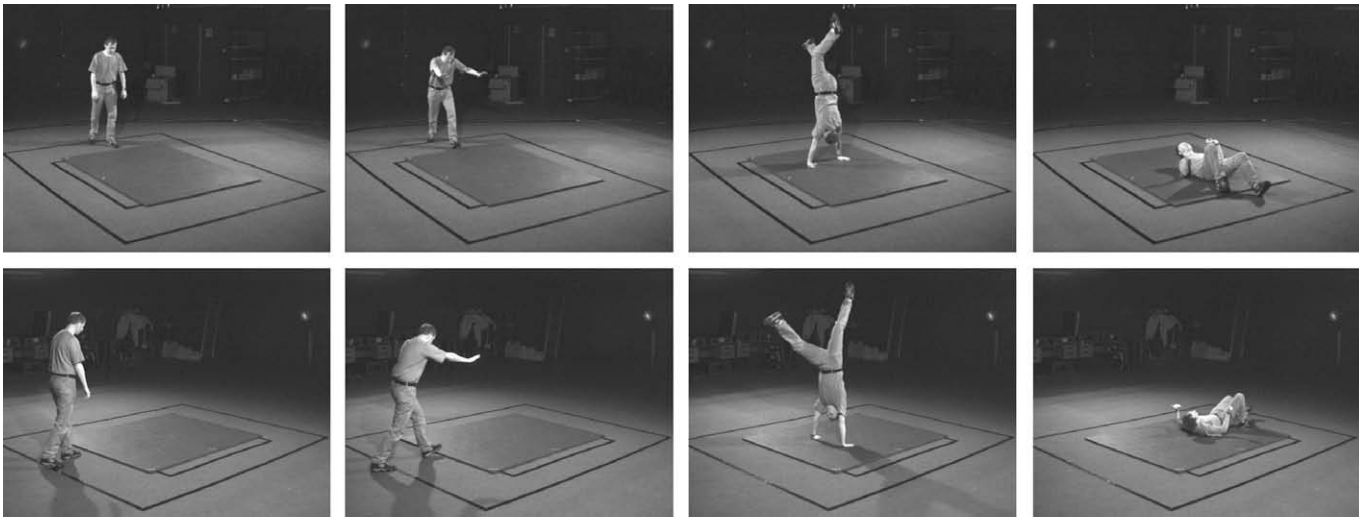

Fig. 9. Handstand sequence as seen from two wide baseline viewpoints.

pared with using the nearest frame. Qualitatively similar results were reported in [24] for a noisy sequence.

We note that there is a tradeoff between the validity of the interpolation method and our confidence in the estimated sub-frame offset. For a slow movement (or high frame rate), the motion between consecutive frames is small such that the assumption of linear motion between frames is most valid. However, the small differences between frames result in a very shallow minimum in the error function such that our uncertainty in the sub-frame offset value is relatively high.

In contrast, faster movements (or slower frame rates) result in larger inter-frame differences in feature locations that are more amenable to interpolation (since they give rise to a more prominent minimum in the error function). However, linear interpolation becomes less valid as the movement between sequences increases.

We also note that faster sequences may exhibit motion blur that increases uncertainty in the feature locations (whether estimated automatically or by a human operator). Although this effect would be reduced for sequences of high-frame rates (or slow movement), we remind the reader that one of the key applications of the method is to synchronize *stock* footage where we have no control over the data capture.

One application where interpolating feature locations is particularly beneficial arises for sequences of different frame rates. In this case, generating synchronized sequences from uninterpolated data results in frames being skipped (in the faster sequence) or duplicated (in the slower sequence). For example, when synchronizing PAL and NTSC sequences (25 and 30 Hz, respectively) using the nearest frame duplicates every fifth frame of the PAL sequence in order to maintain temporal consistency, resulting in 'jerky' motion of the feature locations. In contrast, interpolating feature locations smoothes out these discontinuities resulting in a more aesthetically-pleasing motion.

5.5. Recovery of both α and Δ

In the previous experiments, α was fixed at unity such that Δ was the only remaining parameter to be recovered. Under this constraint for affine projection, the algorithm recovered an offset of

$\Delta = 50.07$ frames – an excellent match for the ground truth offset of $\Delta_{gt} = 50$ frames.

For comparison, with α allowed to vary, the affine algorithm recovered similar synchronization parameters of $\alpha = 1.0001$ and $\Delta = 50.05$; the extra degree of freedom allowed the algorithm to reduce the error in Δ although at a cost of increasing the error in α. In general, it is recommended that α be constrained if the true value is known.

In Section 6.3, we demonstrate the synchronization of sequences of different frame rates using NTSC and PAL cameras. An alternative experiment using interleaved frames of different frequency to synthesize a value of $\alpha_{gt} = 1.50$ was also presented for a noisy sequence in [24].

5.6. Performance w.r.t. number of features

In our final experiment with the synthetic sequence, we evaluate the sensitivity of the algorithm to the number of available feature locations. From a total of 14 available features, we synchronize the sequences using sets of $N = 5, 6, 7, \ldots, 14$ features. For each N, we repeat the experiment up to 20 times using random combinations of the available features. We then computed the RMS error in Δ over all trials for each N. This was then repeated using data with noise of $\sigma_n = 2$ pixels to observe any change in performance.

The outcome of the experiment was that the RMS did not typically change with the number of features used. However, the proportion of trials that successfully converged did decrease with the number of features; some trials were unable to recover the correct initial estimate using only five or six features. Adding noise to the data raised the average RMS error to ~ 0.5 frames (see Section 5.3) but did not change the characteristics with respect to the number of features.

These results suggest that most sets of 5 or more points on a human body are sufficiently non-rigid to recover synchronization accurately. However, increasing the number of features increases the chances of converging to the correct solution. Furthermore, a larger feature set also provides some robustness in the presence of occlusion since not all points are required at every frame.

6. Real examples

6.1. Running sequence

We continue with a real running sequence (Fig. 8) that was used in previous work [24,25] to evaluate the algorithm performance. The sequence pair was captured using two calibrated cameras, hardware-synchronized at 60 Hz, for a quantitative ground truth comparison of recovered synchronization parameters. The sequences were manually offset by 30 frames to give ground truth values of $\alpha_{gt} = 1$ and $\Delta_{gt} = 30$. The locations of 13 joints (shoulders, elbows, wrists, hips, knees, ankles and midriff) were hand-labelled in each frame of the sequences. These feature locations were then projected onto their corresponding epipolar lines (obtained from the known calibration of the cameras) to reduce the noise in the feature locations. However, the error component parallel to the epipolar lines remained.

With α constrained at its known value of 1, an offset of $\Delta = 29.96$ was recovered by the affine algorithm, compared with its true value $\Delta_{gt} = 30$. In comparison, the perspective algorithm recovered an effectively zero-error estimate of $\Delta_{gt} = 30$ frames, since the feature locations had been projected onto their epipolar lines.[3] Allowing α to deviate from its true value resulted recovered values of $\Delta = 29.91$ and $\Delta = 30.00001$ for the affine and perspective models, respectively. Plots corresponding to this sequence are shown in Figs. 2–4.

6.2. Handstand sequence

The algorithm relies upon the motion of the subject being non-rigid, otherwise *all* frames are consistent throughout the sequence and the method is not valid. However, rigid motion of the human body may occasionally occur during certain actions where the body assumes an approximately fixed pose for extended periods of time. We show this to be the case for a handstand sequence of 180 frames (Fig. 9), also captured using synchronized cameras and manually offset by 30 frames. These points were not projected onto their epipolar lines and therefore contain more noise than the previous running sequence.

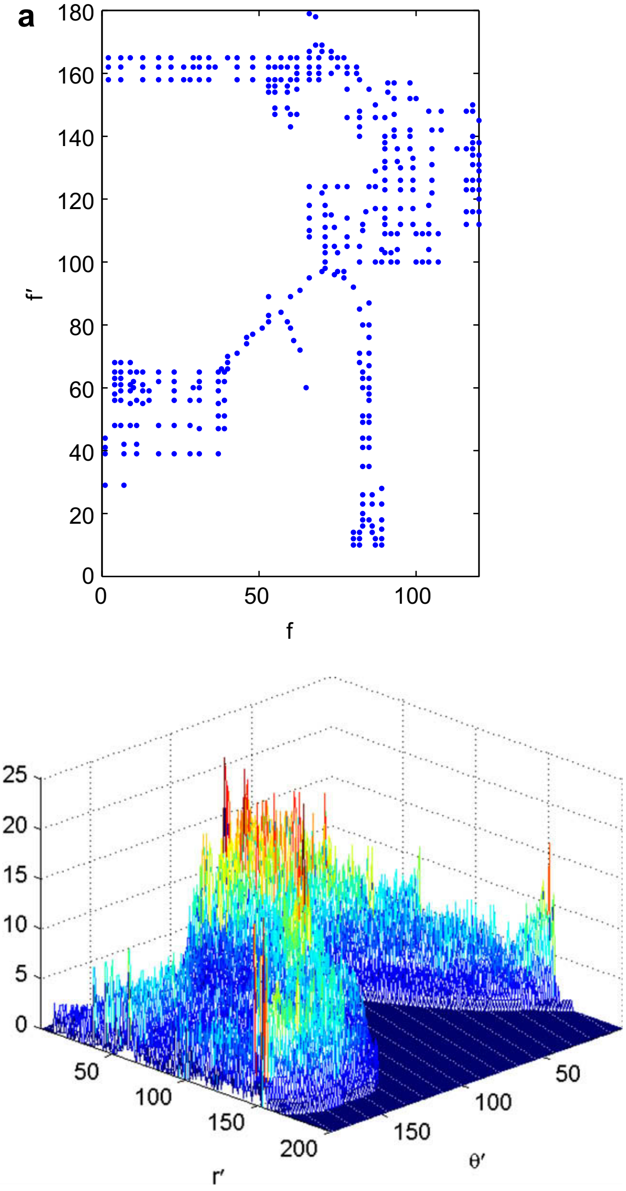

Fig. 10. (a) Recovered correspondences for the handstand sequence and (b) the corresponding Hough accumulator. Compared with Fig. 4, we see no single dominant peak and considerable support for outlying alignment estimates.

[3] Note that only the component of the error that is *normal* to the estimated epipolar line is measurable (and can therefore contribute to the reprojection error minimized by the alignment algorithm) since all points along the epipolar line satisfy the epipolar constraints equally well.

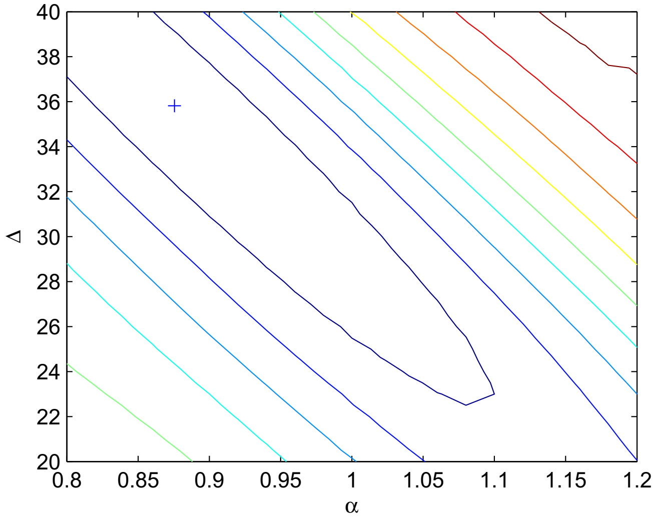

Fig. 11. Contour plot of $C_2(\alpha, \Delta)$ for the handstand sequence. It can be seen that the cost surface is relatively flat compared with Fig. 2b for the running sequence. Furthermore, the minimum of the cost surface appears to be some distance from the true value ($\alpha_{gt} = 1, \Delta gt = 30$).

Fig. 10a shows the putative frame correspondences recovered by the affine algorithm where the underlying linear relationship is clear only for a short period during the middle of the sequence (when the legs undergo a 'scissors' motion). We also observe blocks of corresponding frames suggesting that structure was approximately for extended intervals of time. Fig. 10b shows the corresponding Hough accumulator where we observe a cluster of peaks around the correct solution and many outlying peaks corresponding to spurious estimates.

With α constrained at unity, the 1D Hough accumulator proposed an initial solution of $\Delta = 28$. This was then refined to $\Delta = 28.52$ following non-linear optimization. With α allowed to vary, initial values of $\alpha = 0.900$ and $\Delta = 33.52$ were selected from the 2D Hough array. However, non-linear optimization of these values led to a divergence from the true solution to give final estimates of $\alpha = 0.8755$ and $\Delta = 35.82$.

Fig. 11 shows the cost surface, $C_2(\alpha, \Delta)$, where this local minimum is located some distance from the correct solution. Note that this cost surface is relatively flat, compared with Fig. 2b for the running sequence of identical synchronization parameters, due to the areas of low cost at the extremes of the sequence where the body was almost rigid.

Furthermore, the non-rigid motion that constrains the synchronization occurs in the middle of the sequence. Therefore, any small changes in α to reduce the error must leave the central portion of the line almost unchanged. The only way to achieve this is by applying a large change in Δ to compensate.

We note that one way to increase non-rigidity in the scene would be to include static background points in the feature set. However, since background points can be difficult to match between cameras we do not consider this to be a particularly desirable solution. In any case, rigid human motion is not commonplace and we include this sequence as an exception to the rule. It is also worth noting that in this case the perspective projection model actually performed worse than the affine model for this sequence. The reason for this, however, is unclear.

6.3. Juggling sequence

For our final sequence using the affine camera model, we demonstrate the method on a juggling sequence (Fig. 12) captured using two wide baseline cameras that were neither synchronized nor calibrated. In particular, one sequence was captured using an NTSC digital camera and consisted of 150 colour frames at 30 Hz with a resolution of 320 × 240 pixels. The other sequence, captured with a PAL analogue camera, contained 250 greyscale frames at 25 Hz with a resolution of 720 × 576 pixels. Corresponding feature locations on the upper body, head and juggling balls were again marked manually. We note, however, that this presents a situation where automatic tracking could easily provide feature locations that could be used for synchronization.

Fig. 13a shows the recovered frame correspondences where we observe several distinct parallel bands due to the periodicity of the juggling motion. These are observed as multiple peaks in the Hough accumulator shown in Fig. 13b and multiple minima in the cost function (Fig. 14). From the known frame rates, we computed $\alpha_{gt} = 25/30 \approx 0.833$ and estimated that $\Delta_{gt} \approx 115$ by inspec-

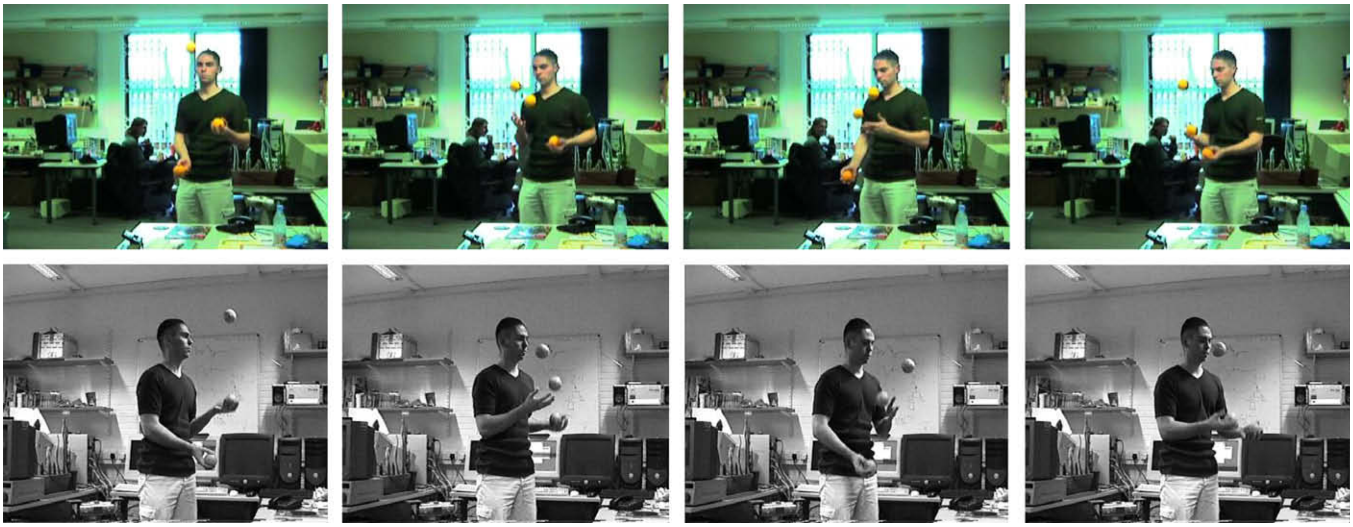

Fig. 12. Juggling sequence as seen from two wide baseline viewpoints.

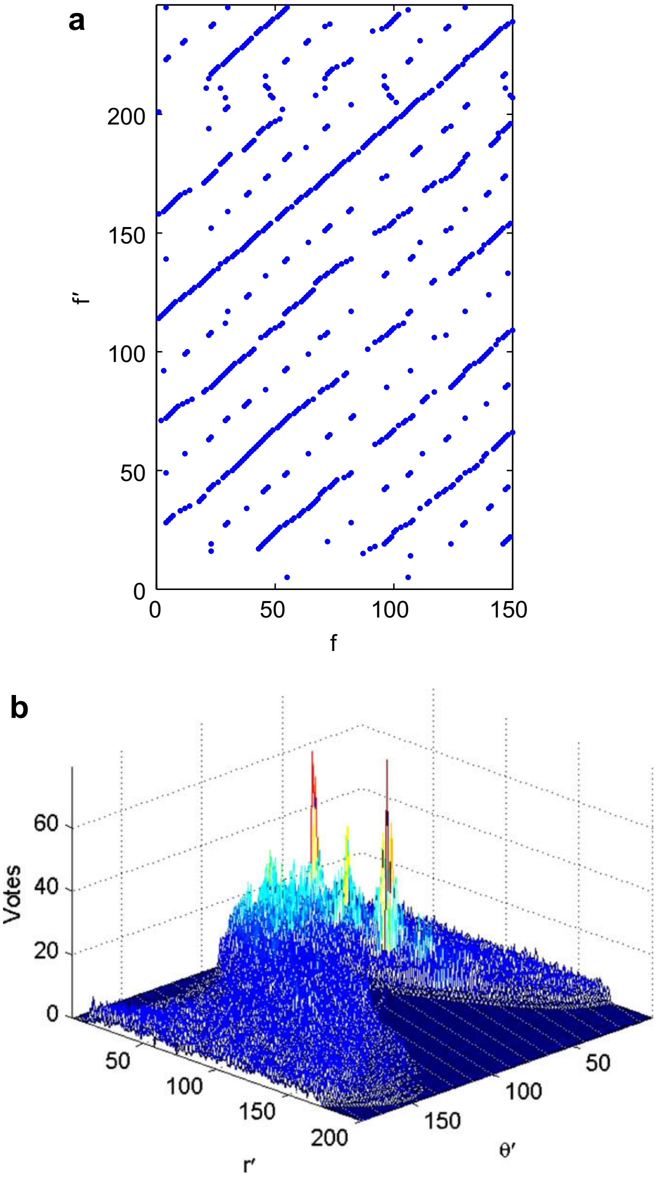

Fig. 13. (a) Recovered correspondences for the juggling sequences and (b) the corresponding Hough accumulator. Note the presence of multiple peaks in the accumulator array due to the periodicity of the juggling motion.

tion. The recovered values of $\alpha = 0.8371$ and $\Delta = 113.60$ closely agreed with these estimates.

Although this would suggest an error in Δ of over one frame, we remind the reader that the 'ground truth' value in this case was estimated by inspection. Therefore, it is more likely that it is this *manually* estimated offset, Δ_{gt}, that is in error rather than the value computed from the data. The purpose of this experiment was simply to show that the algorithm recovered a value of Δ that roughly agreed with our inspection.

6.4. Pins sequence

To finish, we briefly demonstrate the homography model approach using a sequence pair of point features moving independently in a plane, captured using two cameras at approximately 12.5 and 8 Hz. The sequences, shown in Fig. 15, capture map pins moving on a flat surface under the influence of a desk

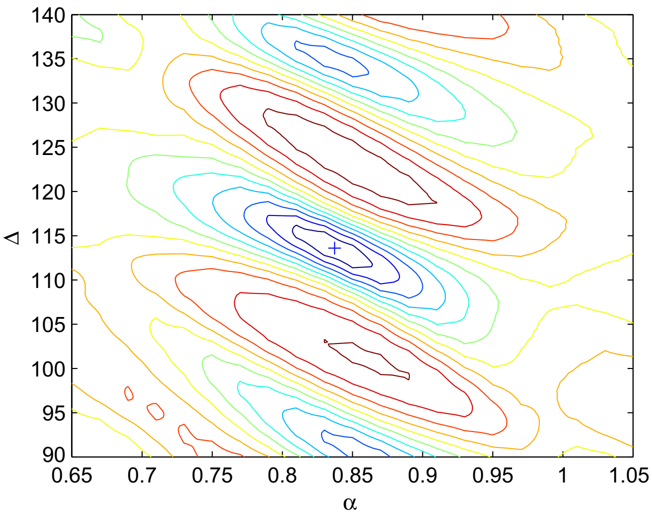

Fig. 14. Contour plot of $C_2(\alpha, \Delta)$ for the juggling sequence. It can be seen that the cost surface has multiple minima due to the periodic motion.

fan. A crude feature tracker was then implemented to recover feature tracks automatically. Although many tracks were corrupted by noise and tracking error, 13 clean tracks were matched by hand.

The recovered frame correspondences and corresponding Hough accumulator are shown in Fig. 16 where very few spurious minima are apparent. The cluster of minima in the lower left corner of Fig. 16a corresponds to the beginning of the sequence, where the pins were static, such that structure was inherently 'consistent'. The true synchronization parameter values were estimated, from the known frame rates and by inspection, as $\alpha_{gt} \approx 0.64$ and $\Delta_{gt} \approx 16$. These values correspond closely to the recovered values of $\alpha = 0.6118$ and $\Delta = 13.50$, demonstrating the effectiveness of the method. As in the previous example, the 'ground truth' value of Δ_{gt} was estimated by inspection and is therefore likely to be uncertain.

There are several reasons for the high performance on this sequence. First, we note that the uncertainty is much smaller for the pins since they are surface features and can be tracked with high accuracy, in contrast to human joint locations that are hidden beneath muscle tissue. Second, the pins were known to move in a plane such that our assumption of corresponding frames being related by a homography was correct, unlike the affine case where perspective effects introduced error into the system. Third, the pins did not assume characteristic configurations, unlike the human body where motion is often periodic. This resulted in far fewer ambiguous matches. Finally, each point feature provides two constraints for cameras related by a homography compared with one constraint each for affine and perspective projection models.

7. Calibration

We note that affine structure and motion from corresponding frames is available almost for free, simply by using the SVD to factorize **W**. However, in order to measure useful kinematic quantities (e.g., joint angles), it is first necessary to remove scaling and skew of the axes by 'upgrading' to a Euclidean coordinate frame. Applying such a calibration method [26,27] to the juggling sequence results in the structure shown in Fig. 17.

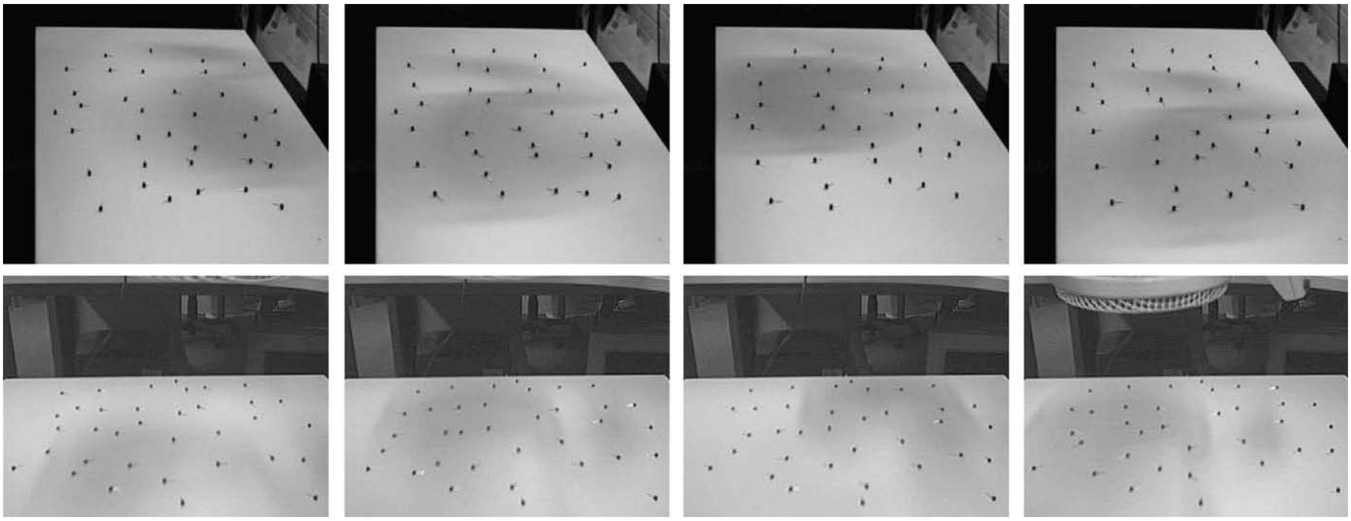

Fig. 15. Pins sequence as seen from two wide baseline viewpoints.

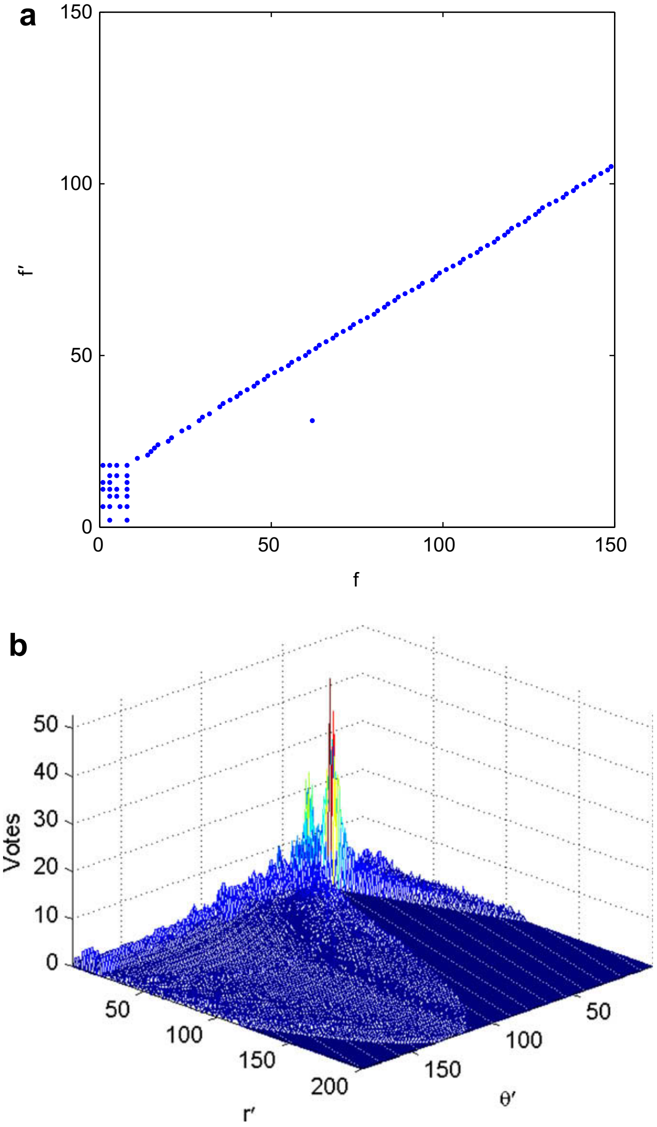

Fig. 16. (a) Recovered frame correspondences and (b) corresponding Hough accumulator with dominant peak. Note that the correspondences and resulting Hough accumulator are considerably more 'clean' than in other cases.

8. Conclusion

This paper has presented a method for synchronizing two sequences of non-rigid motion, demonstrated for human motion in particular. A unified framework was presented that demonstrated the applicability of rank constraints for homography, perspective and affine projection models. This was employed within a computationally efficient algorithm that estimated synchronization parameters to *sub-frame* accuracy for sequences of *unknown and differing frame rates*. A quantitative analysis was undertaken using synthetic data and the method was further demonstrated using real sequences. It was shown that the method could recover synchronization accurately even for sequences of a low frame rate with a relatively large offset. Interpolation was shown to reduce reprojection errors by a factor of 2–3. Sequences of different frame rates were also synchronized successfully.

The main limitation of the method is that it relies on two matched sets of points moving non-rigidly in the scene (in contrast to methods that employ dense image information, e.g., [33]). In this work, we addressed this limitation by manually labelling joint centres in sequences of human motion, although some success has recently been achieved in automating this process [6,12,13]. In other applications, it is often possible to track and match features using standard methodologies (e.g., the KLT tracker [45,46] or WSL tracker [47]) before applying the method, as demonstrated in [23,20]. We also note that, although not shown in this work, the method is applicable for sequences where the cameras *move independently of each other* since each pair of frames is treated independently.

Future development of this method will focus primarily on automating feature tracking within each sequence and feature matching between sequences. In particular, we note that the rank constraints not only indicate temporal alignment between sets of matched points but could potentially be exploited to indicate spatial alignment: correctly matched sets of points result in a lower reprojection error following factorization than incorrectly matched sets. Factorization has also been applied to lines and planes [48], suggesting that these features could be useful for synchronization within a similar framework.

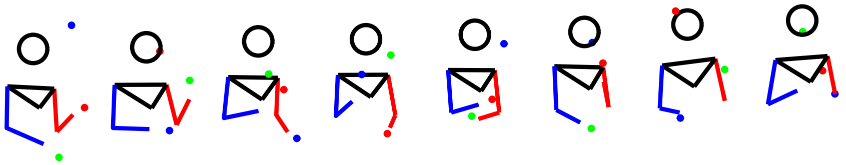

Fig. 17. Euclidean reconstructions from juggling sequence.

Acknowledgments

This project was funded by the Engineering and Physical Sciences Research Council (EPSRC). Image sequences were provided by Oxford Metrics. Data used in these experiments was obtained from mocap.cs.cmu.edu and created with funding from NSF EIA-0196217. We also gratefully acknowledge the constructive comments of our reviewers.

Appendix A. Minimizing $\|\mathbf{Ax}\|^2$ w.r.t. x

Our general aim is to find the vector, $\mathbf{x} = \hat{\mathbf{x}}$, that minimizes $\|\mathbf{Ax}\|^2$ under a scaling constraint that $\hat{\mathbf{x}}^T\hat{\mathbf{x}} = 1$. We begin by computing the SVD of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{U\Sigma V}^T \tag{A.1}$$

and defining

$$\mathbf{y} = \mathbf{V}^T\mathbf{x}. \tag{A.2}$$

Therefore,

$$\|\mathbf{Ax}\|^2 = (\mathbf{Ax})^T(\mathbf{Ax}) \tag{A.3}$$

$$= \mathbf{x}^T\mathbf{V\Sigma}^T\mathbf{U}^T\mathbf{U\Sigma V}^T\mathbf{x} \tag{A.4}$$

$$= \mathbf{y}^T\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U\Sigma y} \tag{A.5}$$

$$= \mathbf{y}^T\mathbf{\Sigma}^2\mathbf{y} \tag{A.6}$$

$$= \sum_{i=1}^{r} y_i^2\sigma_i^2, \tag{A.7}$$

where $r = rank(\mathbf{A})$. It is then straightforward to show that this is minimized by $y_r = 1, y_{i\neq r} = 0$ if the singular values are arranged in descending order of magnitude such that σ_r has the smallest value. This the corresponds to setting $\hat{\mathbf{x}}$ equal to the right singular vector that corresponds to σ_r, as stated.

Appendix B. Normalizing with respect to translation reduces rank by 1

Our aim is to show that translating a set of points such that their centroid lies at the origin results in a smaller matrix that has a smaller rank. More specifically, if

$$\mathbf{M}_A\mathbf{a} = \begin{bmatrix} x_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & x'_N & y'_N & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \mathbf{0} \tag{B.1}$$

then

$$a_1x_i + a_2y_i + a_3x'_i + a_4y'_i + a_5 = 0 \quad \forall i, \tag{B.2}$$

$$\Rightarrow \frac{1}{N}\sum_{i=1}^{N} a_1x_i + a_2y_i + a_3x'_i + a_4y'_i + a_5 = 0, \tag{B.3}$$

$$\Rightarrow a_1\bar{x} + a_2\bar{y} + a_3\bar{x}' + a_4\bar{y}' + a_5 = 0. \tag{B.4}$$

We can also write

$$\begin{bmatrix} x_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & x'_N & y'_N & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0}, \tag{B.5}$$

where

$$\mathbf{s}_1 = \begin{bmatrix} x_1-\bar{x} & y_1-\bar{y} & x'_1-\bar{x}' & y'_1-\bar{y}' & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N-\bar{x} & y_N-\bar{y} & x'_N-\bar{x}' & y'_N-\bar{y}' & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \tag{B.6}$$

$$= \begin{bmatrix} x_1-\bar{x} & y_1-\bar{y} & x'_1-\bar{x}' & y'_1-\bar{y}' \\ \vdots & \vdots & \vdots & \vdots \\ x_N-\bar{x} & y_N-\bar{y} & x'_N-\bar{x}' & y'_N-\bar{y}' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{B.7}$$

and

$$\mathbf{s}_2 = \begin{bmatrix} \bar{x} & \bar{y} & \bar{x}' & \bar{y}' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{x} & \bar{y} & \bar{x}' & \bar{y}' & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \mathbf{0} \tag{B.8}$$

by (B.4). Therefore

$$\begin{bmatrix} x_1-\bar{x} & y_1-\bar{y} & x'_1-\bar{x}' & y'_1-\bar{y}' \\ \vdots & \vdots & \vdots & \vdots \\ x_N-\bar{x} & y_N-\bar{y} & x'_N-\bar{x}' & y'_N-\bar{y}' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \mathbf{0} \tag{B.9}$$

as stated.

Appendix C. The relationship between reprojection error and singular values

In the case of affine projection we consider the measurements, $\mathbf{W}$, to be the sum of our reprojected structure estimate, $\mathbf{PZ}$, and some measurement error, $\widetilde{\mathbf{W}}$. Therefore, using the SVD again gives

$$\widetilde{\mathbf{W}} = \mathbf{W} - \mathbf{PZ} \tag{C.1}$$

$$= \sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T - \sum_{i=1}^{3}\sigma_i\mathbf{u}_i\mathbf{v}_i^T \tag{C.2}$$

$$= \sum_{i=4}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T, \tag{C.3}$$

where r is the actual rank of $\mathbf{W}$, σ_i is the ith singular value of $\mathbf{W}$, and $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith left and right singular vectors of $\mathbf{W}$, respectively. Our aim is to minimize the sum of squared reprojection errors, E, where

$$E = \|\widetilde{\mathbf{W}}\|_F^2 = trace(\widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}}) = \sum_j \lambda_j \tag{C.4}$$

and λ_j are the eigenvalues of $\widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}}$. By definition the singular values of $\widetilde{\mathbf{W}}$ are the square roots of the eigenvalues, λ_j. From the SVD of $\widetilde{\mathbf{W}}$ (Eq. C.3), it can be shown that

$$E = \sum_{i=4}^{r} \sigma_i^2, \tag{C.5}$$

i.e., the sum of the squared singular values is equal to the sum of squared reprojection errors and is therefore a suitable metric for structural consistency between two frames.

References

[1] Vicon motion capture solutions, Online specifications, <http://www.vicon.com/>.

[2] T.B. Moeslund, A. Hilton, V. Kruger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2–3) (2006) 90–126.

[3] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 126–133.

[4] D. Gavrila, L. Davis, 3-D model-based tracking of humans in action: a multi-view approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 73–80.

[5] I. Kakadiaris, D. Metaxas, Model-based estimation of 3D human motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1453–1459.

[6] J. Sullivan, S. Carlsson, Recognizing and tracking human action, in: Proceedings of the European Conference on Computer Vision, vol. 1, Springer LNCS 2350, 2002, pp. 629–644.

[7] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1372–1384.

[8] G. Shakhnarovich, P. Viola, T. Darrell, Fast pose estimation with parameter sensitive hashing, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 750–759.

[9] R. Rosales, S. Sclaroff, Inferring body pose without tracking body parts, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 721–727.

[10] A. Agarwal, B. Triggs, Recovering 3D human pose from monocular images, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 1–15.

[11] C. Sminchisescu, A. Kanaujia, D. Metaxas, BM^{3}E: discrimintative density propagation for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 2030–2044.

[12] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, International Journal of Compute Vision 61 (1) (2005) 55–79.

[13] D. Ramanan, D.A. Forsyth, A. Zisserman, Tracking people by learning their appearance, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 65–81.

[14] R. Ronfard, C. Schmid, B. Triggs, Learning to parse pictures of people, in: Proceedings of the European Conference on Computer Vision, vol. 4, Springer LNCS 2353, 2002, pp. 700–714.

[15] S. Ju, M. Black, Y. Yacoob, Cardboard people: a parameterized model of articulated image motion, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 1996, pp. 38–44.

[16] D. Morris, J. Rehg, Singularity analysis for articulated object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 289–297.

[17] A. Agarwal, B. Triggs, Tracking articulated motion using a mixture of autoregressive models, in: Proceedings of the European Conference on Computer Vision, Springer LNCS 3023, 2004, pp. 54–65.

[18] C. Sminchisescu, B. Triggs, Estimating articulated human motion with covariance scaled sampling, International Journal of Robotics Research 22 (6) (2003) 371–393.

[19] H. Sidenbladh, M. Black, D. Fleet, Stochastic tracking of 3D human figures using 2D image motion, in: Proceedings of the European Conference on Computer Vision, vol. 2, Springer LNCS 1843, 2000, pp. 702–718.

[20] R. Urtasun, D.J. Fleet, A. Hertzmann, P. Fua, Priors for people tracking from small training sets, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 403–410.

[21] I. Reid, A. Zisserman, Goal-directed video metrology, in: Proceedings of the European Conference on Computer Vision, vol. 2, Springer LNCS 1065, 1996, pp. 647–658.

[22] Y. Caspi, D. Simakov, M. Irani, Feature-based sequence-to-sequence matching, International Journal of Computer Vision 68 (1) (2006) 53–64.

[23] T. Tuytelaars, L.V. Gool, Synchronizing video sequences, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 762–768.

[24] P. Tresadern, I. Reid, Synchronizing image sequences of non-rigid objects, in: Proceedings of the British Machine Vision Conference, vol. 2, 2003, pp. 629–638.

[25] P. Tresadern, I. Reid, Uncalibrated and unsynchronized human motion capture: a stereo factorization approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 128–134.

[26] D. Liebowitz, S. Carlsson, Uncalibrated motion capture exploiting articulated structure constraints, International Journal of Computer Vision 51 (3) (2003) 171–187.

[27] P.A. Tresadern, I.D. Reid, Camera calibration from human motion, Image and Vision Computing 26 (6) (2008) 851–862.

[28] D. Pooley, M. Brooks, A. van den Hengel, W. Chojnacki, A voting scheme for estimating the synchrony of moving-camera videos, in: Proceedings of International Conference on Image Processing, vol. 1, 2003, pp. 413–416.

[29] C. Zhou, H. Tao, Dynamic depth recovery from unsynchronized video streams, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 351–358.

[30] R.L. Carceroni, F.L.C. Padua, G.A.M.R. Santos, K.N. Kutulakos, Linear sequence-to-sequence alignment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 746–753.

[31] G. Stein, Tracking from multiple view points: self-calibration of space and time, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 1999, pp. 521–527.

[32] Y. Caspi, M. Irani, Aligning non-overlapping sequences, International Journal of Computer Vision 48 (1) (2002) 39–51.

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

[34] C. Lei, Y.-H. Yang, Tri-focal tensor-based multiple video synchronization with subframe optimization, IEEE Transactions on Image Processing 15 (9) (2006) 2473–2480.

[35] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, International Journal of Computer Vision 50 (2) (2002) 203–226.

[36] L. Wolf, A. Zomet, Correspondence-free synchronization and reconstruction in a non-rigid scene, in: Proceedings of the Workshop on Vision and Modelling of Dynamic Scenes, 2002.

[37] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization approach, International Journal of Computer Vision 9 (2) (1992) 137–154.

[38] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, MA, 2000. ISBN: 0521540518.

[39] H.C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature 293 (1981) 133–135.

[40] R. Hartley, In defense of the eight-point algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (6) (1997) 580–593.

[41] I. Reid, D.W. Murray, Active tracking of foveated feature clusters using affine structure, International Journal of Computer Vision 18 (1) (1996) 41–60.

[42] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Graphics and Image Processing 24 (6) (1981) 381–395.

[43] D. Ballard, C. Brown, Computer Vision, Prentice-Hall, 1982, ISBN 0131653164.

[44] P.H.S. Torr, D.W. Murray, The development and comparison of robust methods for estimating the fundamental matrix, International Journal of Computer Vision 24 (3) (1997) 271–300.

[45] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proceedings of the Image Understanding Workshop, 1981.

[46] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[47] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1296–1311.

[48] D. Morris, T. Kanade, A unified factorization algorithm for points, line segments and planes with uncertainty models, in: Proceedings of the IEEE International Conference on Compter Vision, 1998, pp. 696–702.