Synchronization of User-Generated Videos through Trajectory Correspondence and a Refinement Procedure

Sukrit Shankar
Machine Intelligence
Laboratory, Cambridge
University
ss965@cam.ac.uk

Joan Lasenby
Communications and Signal
Processing Group, Cambridge
University
jl221@hermes.cam.ac.uk

Anil Kokaram
Youtube, Google Inc. &
Sigmedia Group, Trinity
College, Dublin
anilkokaram@google.com

ABSTRACT

Temporal synchronization of multiple video recordings of the same dynamic event is a critical task in many computer vision applications e.g. novel view synthesis and 3D reconstruction. Typically this information is implied through the time-stamp information embedded in the video streams. User-generated videos shot using consumer grade equipment do not contain this information; hence, there is a need to temporally synchronize signals using the visual information itself. Previous work in this area has either assumed good quality data with relatively simple dynamic content or the availability of precise camera geometry.

Our first contribution is a synchronization technique which tries to establish correspondence between feature trajectories across views in a novel way, and specifically targets the kind of complex content found in consumer generated sports recordings, without assuming precise knowledge of fundamental matrices or homographies. We evaluate performance using a number of real video recordings and show that our method is able to synchronize to within 1 sec, which is significantly better than previous approaches.

Our second contribution is a robust and unsupervised view-invariant activity recognition descriptor that exploits recurrence plot theory on spatial tiles. The descriptor is individually shown to better characterize the activities from different views under occlusions than state-of-the-art approaches. We combine this descriptor with our proposed synchronization method and show that it can further refine the synchronization index.

Categories and Subject Descriptors

I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - Motion, Video Analysis; I.4.8 [Pattern Recognition]: Applications - Computer Vision, Video Production

General Terms

Design, Experimentation, Media Production

Keywords

Event detection, Synchronization, Optical Flow

1. INTRODUCTION

In all computer vision applications relying on multi-view geometry combined with image sequence analysis, temporal synchronization of the different recordings is very important. Some prominent examples are 3D reconstruction and visualisation [25], video mosaicing [2], robust multi-view video-based surveillance [37], efficient search of video portions within a database of videos and graphics modification in athletics broadcasts [19]. Video synchronization is also necessary for the task of stitching of user-generated videos [28]. As of now, such stitching methods heavily rely on manual synchronization of videos; thus, even a reasonably accurate automatic synchronization of real recordings is helpful. The extraction of geometry information is usually contingent on the fact that each frame across several recordings has been taken at the same time. This information is usually embedded in the video stream by way of time-stamps or implied in the use of a “genlock” signal ensuring that the recording timebase is the same for each of the cameras used. However, most consumer grade equipment does not contain time-stamp information useful enough to be used for synchronisation. Hence it is difficult to apply the geometric tools of computer vision to multi-view footage recorded from sources such as mobile camera phones.

This problem has been recognized in commercial software such as Adobe Premier Pro and Apple Final Cut Pro; packages which provide a post-processing manual alignment facility. Manual synchronization is time consuming and clearly does not scale to large collections. If the recordings share the same audio environment, it is also possible to use audio correlation to automatically synchronise the signals [10]. However, in the case of sports footage in particular, shared audio environments are not regularly available. Hence exploiting correlation in visual features is the only way to achieve synchronization in such environments.

1.1 Video Synchronization - Problems and Related Work

Previous work for synchronizing videos has generally been centered around extraction of some time varying fingerprint from each video sequence. The problem of temporal synchronization is then solved by estimating the temporal offset that maximizes the correlation between these fingerprints. Several synchronization methods have considered moving cameras viewing a static scene [26, 31] or a scene with relatively little motion [13, 23]. The main idea in these works is to find both the spatial and temporal correspondence between point features at frames taken in all possible time shifts between the two video streams. Such approaches are vulnerable to correspondence ambiguities and do not cope with complex scene

dynamics like those common in user-generated videos.

Caspi et al. [2], were the first to exploit motion trajectories as a fingerprint. They assume that a pair of trajectories in two views can be associated with the movement of the same feature in the 3D world. By estimating the fundamental matrix mapping each point in one trajectory onto the other, they generated a matching error based on the distance between the epipolar line in one image and the nearest point in the trajectory. That matching error is minimised when the temporal offset between the two sequences is correct. Improved video synchronization approaches based on motion trajectory fingerprints have been proposed by [3, 4, 25, 30]. Although such methods are relatively robust for the cases where the moving entities are fewer in number, they do not prove robust for scenes containing multiple people, primarily because they assume that the extracted trajectories are always correct. Such an assumption is not valid in real recordings.

There have been attempts at overcoming the problem of 3D object motion during trajectory extraction. The research of [35, 13] uses a three-camera system and trifocal tensors to model 3D motion. Another direction of work has assumed an affine projection and used a linear combination approach in order to avoid exact point correspondence [36, 30]. All these methods rely heavily on accurate knowledge of homographies and/or the fundamental matrices. Although researchers have tried to estimate the homography and the fundamental matrix simultaneously [5], a precise estimation is not possible for user-generated videos.

Pundik and Moses [18] took an alternative approach for video synchronization, where they avoided the use of specific feature trajectories. They extracted temporal motion signals using epipolar lines and such temporal signals were spatially matched using the fundamental matrix. Their approach works well for crowd videos but is very sensitive to the accuracy of the fundamental matrix.

Since the most salient portions of the videos are normally the regions where activities are performed by people, some researchers have used activity descriptors for temporal alignment of videos. Dexter et al. [8] proposed self-similarity matrices for describing motions under different viewing angles, and then used these descriptors for video synchronization. However, their work is robust only for simple actions without occlusions. The work of [14] has tried to extend the self-similarity matrices for characterizing group activities; they assume very precise knowledge of 2D trajectories of humans/objects. The work of [9] estimates poses of the players in sports videos for aiding the task of video synchronization. However, their method relies on precise silhouette extraction of the various people involved in the video and thus is very limited.

All of these previous techniques do not cope well with consumer grade recordings and especially sports video content recorded by amateurs because (a) the picture quality ensures that the fundamental matrix/homography cannot be estimated with accuracy and (b) association of trajectories to the same movement in the 3D world is not a trivial problem. We propose an approach which achieves correspondences by selecting most plausible correct trajectories (while taking into account their inherent 3D motion) without any assumption of a precise fundamental matrix, homography, poses or activity descriptors; and utilizes these correspondences in a novel way to infer the time alignment index. This method helps to achieve synchronization for user-generated sports videos to within 1 sec, which is significantly better than the state-of-the-art approaches. Once

synchronization within a close approximation is achieved, one can apply some refinement procedures to get closer to the correct synchronization index.

1.2 View-invariant Activity Recognition - Problems and Related Work

Using pose estimation or activity recognition methods on motion-segmented regions containing human subjects cannot generally be applied to the task of synchronizing videos of significant duration. Such methods are only robust for very cleanly separated motion regions and in user-generated videos (like that of sports events), the presence of multiple subjects makes the extraction of such motion regions difficult. The presence of occlusions, etc. also makes it difficult to apply pose estimation and activity recognition descriptors to anything other than the clips of small temporal spans. **Thus, for an entire complex video (such as those in our dataset), it is wise to make use of these techniques as refinement procedures only, when we have already obtained an approximate range of synchronization indices.**

For the videos shot from a far-off distance as in our dataset (containing recordings of sports events), it is almost impossible to automatically estimate the poses of human subjects. Some researchers [9] have tried to estimate poses of subjects in sports events like hockey; however, their methods are far from automatic and rely heavily on manual extraction of human silhouettes for all frames of the video. We thus resort to activity descriptor based methods. Some recent works on activity recognition [21, 32] present state-of-the-art results on datasets like Youtube, Hollywood and HMDB [12]. However, these methods require a good amount of training data for efficient classification of feature descriptors. Also, the techniques are either trajectory-based or are based on high-level representation using an ensemble of Gabor filters, and thus cannot adapt to cross-view activities. For refining a synchronization index across videos, one needs view-invariant activity descriptors which can be used in an unsupervised manner as far as possible. Also, the descriptors should be somewhat resistant to occlusions, since even in a reasonably accurate extracted motion region, occlusions due to people crossing each other are hard to avoid.

In pursuit of developing algorithms for view-invariant activity descriptors, researchers have tried to store templates from several canonical views as done in [1] and interpolate across the stored views as proposed by [7]. This approach however is not scalable since one does not know how many views to consider as canonical. Some other state-of-the-art feature descriptors for encoding actions/activities in the presence of occlusions under different view-points exist [34], while some action recognition procedures like [27, 22, 17] are comparable to the state-of-the-art procedures. However, none of them is automatic and most of them do not consider multiple views under occlusions. The self-similarity descriptor (SSM) of [11] presents an unsupervised solution to view-invariant activity recognition and has been used for automatic synchronization of videos [8]. However, the results are promising only for videos with very cleanly extracted activities and in the absence of occlusions.

In this paper, we propose *part-SSMs* which are extension of SSMs on spatial tiles. The extension is not straightforward since matching such spatial tiles requires that the tiles not exhibiting any patterns are adequately discarded in an automatic manner. We show that *part-SSMs* are much more robust to occlusions than SSMs,

and they also help to refine the results with our synchronization approach better than SSMs.

Our major contributions in this paper can now be enumerated as follows:

1. A method that directly addresses the problem of discovery of trajectories which can be associated with the same movement in the 3D world. This is used to achieve an approximate synchronization index (usually within 1 sec).
2. An automatic view-invariant activity descriptor robust to occlusions (part-SSMs) which can be used to further refine the synchronization results.

2. SYNCHRONIZATION APPROACH

This section presents a detailed description of our approach for time alignment of video sequences. Let us consider a pair of unsynchronized sequences V_1 and V_2 . Let the number of frames in the sequences be N_1 and N_2 respectively. The problem is to estimate the frame offset t^* in V_1 with which the first frame of V_2 is temporally aligned. This is under the assumption that the content of V_2 is completely contained within the duration of V_1 , and thus we also have $N_1 > N_2$.

The main idea behind our approach is that for a moving point in V_1 , its 2D trajectory can be related by a warping function to the trajectory of a point in V_2 ; and that all such corresponding moving points across the two videos will have a maximally similar warping function at the correct time shift. This means that all the trajectories related in the two videos should ideally be related by a common warping function at the correct time shift. At incorrect time shifts, the trajectories cannot be related by a common warping function, but only by a nearly common one. Such a correspondence takes into account three important dynamical features of 2D trajectories. First, since the trajectories are related by a warping function, the inherent 3D motion is taken into account. Second, the warping functions do not care whether the trajectories are slightly erroneous (due to occlusions) or not. We are just concerned that the correctly related trajectories (across the two videos) should have similar warping functions between them. This assumption is valid since the dynamics present in both video sequences are similar, as the videos are from the same event¹. Third, since correctly related trajectories share a nearly common warping function, all other erroneous trajectory correspondences are neglected.

Each sequence is pre-processed so that every frame is registered to its preceding frame in the sequence. For this, we employ a typical feature based image registration technique using the Scale Invariant Feature Transform (SIFT) [15] method. An affine model is used for registration. This accounts for global or camera motion in each view and thus allows the algorithm to focus on local or foreground motion. In amateur sports footage, we can assume that much of the scene will contain static objects e.g. the ground or the stadium. Hence we estimate an approximate fundamental matrix between

¹Although the dynamics in the two sequences are almost the same when the videos are shot from the same event, there can be slight variations. For example, in one view, two people can occlude each other; while in the other they might not. In such a case, the warping functions for pairs of related trajectories will only be nearly similar. This is handled by our algorithm since the correct temporal alignment is only dependent on the maximally similar warping function, and not on any absolute basis.

the two clips by taking the first 100 frames from each sequence and using a standard method [20] to estimate $\mathbf{F}$ for each of these frames. The approximate fundamental matrix $\hat{\mathbf{F}}$ used for further analysis is taken as the mean of these 100 estimates². To restrict the search for trajectories, we only use the parts of the frame which are indicated as *motion regions*. The details of this procedure are discussed in the following sections, but the result is a mask for each frame which is 1 in regions that will be searched for trajectories and 0 otherwise. The remaining process can be divided into methods for trajectory extraction, matching and synchronization. The details of each of these steps are discussed next.

2.1 Trajectory Extraction

We describe here the procedure for extraction of trajectories from the two videos V_1 and V_2 . For an estimate of the time offset $t; t = 1, \dots, (N_1 - N_2)$ being examined, we first of all discard the first t frames in V_1 . This is because we only need to find the desired index t^* in V_1 which best corresponds to the first frame of V_2 . We then extract approximate motion regions M_1 and M_2 from videos V_1 and V_2 respectively. The motion regions are extracted over a 10 frame interval, i.e. we take an absolute difference of the intensities of the 10^{th} frame with the 1^{st} frame, the 20^{th} frame with the 11^{th} frame, and so on. We then threshold these difference values according to a threshold value $\tau = 0.3$ on a normalized scale of 1. We take the union of these regions to give M_1 and M_2 . Some examples of the extracted motion regions are given in Fig 1(c).

For the motion region M_1 in V_1 , we randomly sample some K feature points. These K points represent a subset of points which were in motion possibly for different intervals of time in video V_1 . We track the 2D trajectories of these K points using the Kanade-Lucas-Tomasi (KLT) tracker [24]. Each of the K sampled SIFT (feature) points is tracked through some subset L of consecutive frames ($L = 50$ in our case) to give a trajectory $P_1^j = \{(h_1, k_1)_j, (h_2, k_2)_j, \dots, (h_L, k_L)_j; j = 1, \dots, K\}$, where (h_n, k_n) is the position of the KLT point in the n^{th} frame of V_1 . We limit the trajectory length to $L = 50$ frames in order to avoid the drift of trajectories with time. This is reasonable since normally a discriminative action in a video does not continue for more than 2 seconds. Given a SIFT (feature) point in V_2 , a trajectory $P_2^r = \{(x_1, y_1)_r, (x_2, y_2)_r, \dots, (x_L, y_L)_r; r = 1, \dots, K\}$ (where (x_n, y_n) is the position of the KLT point in the n^{th} frame of V_2) can be tracked in a similar fashion over L consecutive frames.

For each trajectory P_1^j in V_1 starting at a point $\mathbf{p}$, we use $\hat{\mathbf{F}}$ to find the corresponding point $\mathbf{q}$ in the first frame of V_2 . We achieve this by mapping the SIFT (feature) point $\mathbf{p}$ to a point in the first frame of V_2 along the epipolar line in that frame. The point $\mathbf{q}$ is selected to be visually the same by matching the SIFT feature vector at each point along the line. In an ideal scenario where we can assume $\hat{\mathbf{F}}$ to be precise, matching $\mathbf{p}$ to the corresponding point $\mathbf{q}$ in V_2 suffices. However, since for real recordings (as in our case), $\hat{\mathbf{F}}$ cannot be assumed to be precise, we take A points in V_2 around $\mathbf{q}$ and try to match their trajectories with that of $\mathbf{p}$ in V_1 . For this, we take a 2D Gaussian distribution (with mean at $\mathbf{q}$ and some variance Σ) and sample A points in the region. This notion is illustrated in Fig 1(a) and Fig 1(b). For K trajectories in V_1 and A possible trajectories

²Note that in real environments, an accurate estimation of the fundamental matrix is not guaranteed and thus, the algorithm should be able to operate robustly with an approximate fundamental matrix. This is in contrast to most other methods which generally assume a precise knowledge of fundamental matrices/homographies.

Figure 1: (a) An illustration of sampling a number of points around the region of q to compensate for an approximate Fundamental Matrix $\hat{F}$. (b) The trajectory of the feature point p in the first video is matched to the trajectories of some randomly sampled points in the second video within a 2D Gaussian distributed region around q . (c) Extracted motion regions in the (from top to bottom) cricket, baseball and football videos.

in V_2 for each trajectory in V_1 , we thus have a total of KA possible matches. Also, since A points in V_2 have been taken from a 2D Gaussian distribution, each trajectory match has an associated probability measure $g_{j,r_j}; j = 1, \dots, K; r_j = 1, \dots, A$. We next describe the procedure for matching the extracted trajectories.

2.2 Trajectory Matching

Let us consider two trajectories $P_1^j = \{(h_1, k_1)_j, (h_2, k_2)_j, \dots, (h_L, k_L)_j\}$ and $P_2^r = \{(x_1, y_1)_r, (x_2, y_2)_r, \dots, (x_L, y_L)_r\}$ in V_1 and V_2 respectively, which are to be matched. We first map each of the trajectories onto a common reference point $(0, 0)$. We then put the two trajectories onto a common sampling grid $\{h_1, h_2, \dots, h_L\}$, i.e. P_2^r is transformed to $\hat{P}_2^r = \{(h_1, \hat{k}_1)_r, (h_2, \hat{k}_2)_r, \dots, (h_L, \hat{k}_L)_r\}$ using cubic interpolation. The notion is illustrated in Fig 2. We assume that the trajectories are related by a polynomial of order W , which we call the warping function $\varphi_{j,r}(h_q) = 1 + \sum_{w=1}^W (\theta_w)_{j,r} h_q^w$; $q = 1, \dots, L$. Let the coefficients $(\theta_w)_{j,r}; w = 1, \dots, W$ of the polynomial be represented by $\Theta_{j,r}$ for relating trajectories P_1^j and $\hat{P}_2^r$. Then, we estimate $\Theta_{j,r}$ as

$$\arg \min_{\Theta_{j,r}} \sum_{q=1}^L (k_q \varphi_{j,r}(h_q) - \hat{k}_q)^2 \quad (1)$$

Note that for each trajectory P_1^j in V_1 , we need to match it to A trajectories in V_2 denoted as $P_2^{r_j}; r_j = 1, \dots, A$. Thus, we obtain Θ_{j,r_j} containing KA parameter sets Θ . We then cluster these KA parameter sets using k-means clustering [16] (with 4 clusters). We find the cluster C with the maximum number of points and note its variance as σ_{C_t} . The largest cluster C contains the parameters of the warping function which relates the greatest number of trajectories at a given time shift. The variance σ_{C_t} of the cluster tells us about the similarity of the warping functions which best related the maximum number of trajectories across the two videos. At the correct time shift, one would expect the variance σ_{C_t} to be a minimum, making the trajectories related by a maximally similar warping function.

2.3 Synchronization

We now need to address the fact that we have only considered K random points in M_1 . Since the K points are chosen at random, it is likely that we will miss some points in M_1 whose trajec-

tries exhibit better matches to those in V_2 . Thus, at a given time shift, we iterate by altering a number of feature points until σ_{C_t} becomes constant for two consecutive iterations. This ensures that for a given time shift, we have selected the best feature points whose trajectories can be related by a nearly similar warping function. In formal terms, for a given time shift t , at each iteration i_t , out of KA trajectory correspondences, $\lfloor \alpha KA \rfloor; \alpha = 0.3 (< 1)$ correspondences are altered and a new set of KA trajectory correspondences is estimated. Let the number of points in V_1 associated with the discarded $\lfloor \alpha KA \rfloor$ correspondences be $K_1 \leq \lfloor \alpha K \rfloor$. Then, K_1 new feature points are sampled in M_1 , their trajectories are extracted and matched with those of some points in V_2 according to the procedure described in Sec 2.1 and Sec 2.2. The $\lfloor \alpha KA \rfloor$ correspondences which are discarded at every iteration are the ones which have the lowest associated probabilities $g_{j,r_j}; j = 1, \dots, K; r_j = 1, \dots, A$. By considering the associated probabilities for discarding purposes, we assume that the approximate fundamental matrix $\hat{F}$ is reasonably accurate; and that we can therefore discard those trajectory correspondences which were made between a point p in V_1 and a relatively far away point q_a (from q) in V_2 . For each i_t , the variance of the largest cluster (of parameters of the warping functions) $\sigma_{C_{i_t}}$ is calculated until $|\sigma_{C_{i_t}} - \sigma_{C_{(i+1)_t}}| < \epsilon (= 0.01)$, after which $\sigma_{C_t} = \sigma_{C_{i_t}}$. This process is illustrated in Fig 3. Finally, the correct time alignment index t^* is given by:

$$t^* = \arg \min_{t=1, \dots, (N_1 - N_2)} \sigma_{C_t} \quad (2)$$

Note that for the purpose of establishing correspondence between trajectories, a homography matrix could also have been estimated instead of a warping function. However, a relation through a homography matrix is reliable only when the cameras do not undergo any translation. Due to the complexity of the videos in our dataset, it is difficult to estimate an accurate epipolar geometry, and thus translational motion cannot be completely got rid of. **Thus, in such cases, establishing relations between trajectories with the help of a homography matrix is highly erroneous and often makes the procedure (3)(d)(iv) of Algorithm 1 non-convergent.**

2.4 Edge Map Sampling (EMS):

We now introduce a modification to our procedure so that approximate image geometry can be usefully exploited. To ensure that the most discriminative moving feature points are considered in both

Figure 2: Top Row (Left Column) - An illustration of the common sampling axis of trajectories P_1^j and $\hat{P}_2^r$ for the estimation of the warping function. The red points indicate points on the initially extracted trajectory P_2^r . The warping function $\phi_{j,r}(\mathbf{h})$ gives L values for $\mathbf{h} = \{h_1, h_2, \dots, h_L\}$ which map P_1^j and $\hat{P}_2^r$ according to Eqn (1). **Top Row (Right Column)** - *Edge Map Sampling* - Some points are sampled on the edge map within a motion region. The intuition behind this is that the points along the edges in a motion region provide more discriminative trajectories than otherwise. Here, for a person running in a football match, with the help of edge map, points around the legs and waving hand are sampled which can give more discriminative trajectories than those on the torso. *Best viewed in color.*

Figure 3: (Four) plots showing the scatter of parameters of the warping functions in 2D space (projected from 10D space) for estimating σ_C at a particular t according to Step (3)(d) of Algorithm 1. In each figure, **RED** represents the cluster C with the maximum number of points. The points marked with other colors represent the plausibly erroneous trajectory matches. It can be seen that with an increasing number of iterations, the variance of the largest cluster converges to a minimum to give σ_C . *Best viewed in color.*

videos, we propose the approach of *Edge Map Sampling*. The main idea is that the points along edges in a motion region provide more discriminative trajectories than otherwise. For example, consider a person running and bowling. While the trajectories of the feature points on his torso will all have roughly the same motion (moving forward), the trajectory of the point on his hand depicts the more discriminative motion of bowling. In our case, the edge map is calculated using the Sobel filter. We sample around 20% of the points (out of K points in V_1 and KA points in V_2) on the edge maps of motion regions. An illustration of sampling points around the edge map is given in Fig 2. Algorithm 1 lists the steps of our proposed approach.

3. REFINING WITH PART-SSM

A brief description of the self-similarity descriptors of [11] is given in Algorithm 2. A self similarity structure is essentially a square sized image as shown in Fig 4. Once SSM is formed, a temporal localization procedure is performed as in [11]. Around every point on the diagonal of the SSM structure, a polar grid is formed with bins in it. Usually, the polar grid is divided into 11 bins. Then the HOG-LBP [33] descriptor for each bin are concatenated to form the final descriptor for the SSM. This is depicted in Fig 4.

We propose to use the self-similarity matrices on spatially partitioned regions of the video. For instance, for a video of frame size 200×200 , instead of applying the SSM on the full 200×200 spatial space, SSMs are applied on spatial spaces of size 40×40 , thereby producing 25 SSM structures. These SSM structures then need to be processed for matching and finding correspondence across

Figure 4: Comparison of Global Self Similarity Structures (SSMs) with Partitioned Self Similarity Structures (part-SSMs) for a simple hand waving activity video. (LEFT) The first structure is the plain SSM structure [11] for the entire video. **(CENTRE and RIGHT)** The next two structures are two (out of 24) part-SSMs for the same video. It can be seen that the structures on partitioned spaces exhibit more structured patterns. The axes of the self similarity structures represent time (number of frames in the video).

videos. Note that the proposed spatial partitioning should not be confused with part based models for activity recognition such as [29]. Models as of [29] generally consider partitioned structures of human limbs and joints, and therefore require robust human detection along with rough positions of joints.

The idea behind using spatial partitioning is as follows:

- With spatial partitioning, one can handle occlusions in a much better way, since activity in some occluded portions can still be traced in a discriminative manner.

Algorithm 1 Warping Trajectories (WT) for Video Synchronization

1. Register the two unsynchronized videos V_1 and V_2 , with N_1 and N_2 frames respectively ($N_1 > N_2$).
 2. Estimate the approximate fundamental matrix $\hat{\mathbf{F}}$ between V_1 and V_2 .
 3. For all possible time alignment indices $t = 1, \dots, (N_1 - N_2)$, **do**
 - (a) Shift V_1 by t frames.
 - (b) Find approximate motion regions M_1 and M_2 in the two videos.
 - (c) Sample K feature points randomly within M_1 .
 - (d) Find the trajectories for these K points to give $P_1^j; j = 1, \dots, K$.
 - i. For all trajectories $P_1^j; j = 1, \dots, K$, **do**
 - A. For a trajectory P_1^j starting in point $\mathbf{p}$ in V_1 , use $\hat{\mathbf{F}}$ to find the corresponding point $\mathbf{q}$ in V_2 .
 - B. Take a 2D Gaussian distribution (with mean at $\mathbf{q}$ and variance Σ) and sample A points in the region.
 - C. Track these A points in V_2 to get $P_2^r; r = 1, \dots, A$.
 - D. Estimate the warping function parameters between P_1^j and $P_2^r; r = 1, \dots, A$ to get Θ_{j,r_j} using Eqn (1). Each such correspondence is then associated with a probability g_{j,r_j} .
 - ii. Do clustering over Θ_{j,r_j} and find the largest cluster C with variance σ_{C_i} .
 - iii. Ignore $\lfloor \alpha KA \rfloor$ correspondences in Θ_{j,r_j} with least g_{j,r_j} . Let $\lfloor \alpha KA \rfloor$ correspondences lead to K_1 new feature points in V_1 . Randomly sample K_1 new points in M_1 .
 - iv. Go to Step 3(d) and repeat until σ_{C_i} converges.
 4. Estimate the correct time alignment index t^* via Eqn (2).
-

Figure 5: Procedure for discarding part-SSMs. It can be seen that the SSMs exhibiting some patterns show lower amount of high frequency content, as compared to those which do not show any discernible pattern. The green rectangles signify the SSM patterns that were chosen based on their frequency content. The SSMs are displayed in grayscale mode, just to give a different viewpoint of seeing patterns.

- With spatial partitioning, some of the subtle movements within an activity are also revealed, since the features become localized. However, in a global SSM, the patterns always correspond to the main activities.

Figure 6: TOP ROW - Motion regions labelled manually for two cricket videos approximately synchronized by the WT procedure. For these motion regions, descriptors like part-SSMs are extracted. For comparison purposes, all combinations of possible matches between the part-SSMs of motion regions are considered and the best set of matches is considered for refining the synchronization index. For example, there are two motion regions identified in Video 1 (LEFT), and 3 motion regions identified in Video 2 (RIGHT). After considering all possible matches, the two motion regions of Video 1 that correspond to those of Video 2 are color-coded (Black corresponds to black, and yellow to yellow). BOTTOM ROW - Depiction of temporal localization of an SSM structure (©IEEE 2011 [11]).

Algorithm 2 Self Similarity Matrices [11]

1. Let T be a multi-linear 3D array.
 2. For each of the 2D planes in T , apply a feature descriptor such as Histogram of Oriented Gradients (HOG) [6]. This will give a 2D array $T_{feature}$ of size $f \times L$, where L is the size of the computed HOG descriptor.
 3. Take Euclidean distance of the HOG descriptor of each frame with every frame of the video. This will generate a 2D image whose entries can be described as $d(i, j) = D(T_{feature}(i, :), T_{feature}(j, :)); i = 1 \dots f; j = 1 \dots f$, where D corresponds to the Euclidean distance measure. This 2D image is the Self Similarity Structure of size $f \times f$.
-

The above mentioned notions are depicted in Figure 4. Using the proposed concept of partitioned self similarity structures (part-SSMs), we present a procedure for activity recognition and synchronization in Algorithm 3. The major thing in the proposed algorithm is the discarding of the redundant part-SSMs, which correspond to occlusions. An unsupervised approach for the same is given in Algorithm 4 which is referred to in Algorithm 3. It is seen that the SSM structures corresponding to occluded regions contain mostly the high frequencies as compared to the structures depicting a discernible pattern. The same notion is used in Algorithm 4. Figure 5 depicts the procedure for discarding part-SSMs based on their frequency content. For matching, temporal localization of part-SSMs is performed and HOG-LBP [33] descriptors are calculated for characterizing the patterns in each temporally localized space. This procedure is outlined in Algorithm 5.

We intend to use part-SSMs mainly as a refinement tool for synchronization results obtained using the trajectory correspondence approach. Let the approximate synchronization achieved between

Algorithm 3 Activity Recognition and Synchronization with Partitioned Self Similarity Structures

1. Initial Setup

- (a) Let V_1 and V_2 be the two unsynchronized videos having f_1 and f_2 frames respectively.
- (b) Make two multi-linear arrays T_1 and T_2 out of the individual frames of unsynchronized videos V_1 and V_2 .

2. Make Partitioned Structures

- (a) For each of the frames in T_1 and T_2 , make spatial tiles to create multi-linear arrays $T_{1_{\text{tiled}}}$ and $T_{2_{\text{tiled}}}$, each with N tiles per frame. If no spatial tiling is done, $N = 1$.
- (b) Apply Self Similarity Matrix Transformation to all the tiles in arrays $T_{1_{\text{tiled}}}$ and $T_{2_{\text{tiled}}}$ using Algorithm 2. This will create feature encoded multi-linear arrays for the two videos, $T_{1_{\text{feature}}}$ and $T_{2_{\text{feature}}}$. Note that each of $T_{1_{\text{feature}}}$ and $T_{2_{\text{feature}}}$ will contain N self similarity structures (which are 2D images) of sizes $f_1 \times f_1$ and $f_2 \times f_2$ respectively.

3. Discard Redundant Tiles

- (a) Discard the redundant tiles from each of $T_{1_{\text{feature}}}$ and $T_{2_{\text{feature}}}$ using Algorithm 4. This will give feature encoded multi-linear arrays $D_{1_{\text{feature}}}$ and $D_{2_{\text{feature}}}$ with $N_1; N_1 < N$ and $N_2; N_2 < N$ tiles respectively. These arrays now have all the tiles which correspond to some activity happening in the respective videos, with tiles for all regions having occlusions and lacking motion removed.

4. Merge Tiles

- (a) Take mean of all N_1 tiles of $D_{1_{\text{feature}}}$ to get a $f_1 \times f_1$ self similarity structure S_1 .
- (b) Take mean of all N_2 tiles of $D_{2_{\text{feature}}}$ to get a $f_2 \times f_2$ self similarity structure S_2 .

5. Recognize or Synchronize

- (a) Form a comparison matrix according to Algorithm 5 between S_1 and S_2 to get L_1 and L_2 .
 - (b) For recognition purposes, take Euclidean Distance measure between L_1 and L_2 .
 - (c) For Synchronization purposes, take Euclidean distance of $L_2(1, :)$ with $L_1(j, :); j = 1 \dots f_1$. The desired index for synchronization is the one where the Euclidean distance is the minimum.
-

the two videos V_1 and V_2 be such that the first frame of V_2 aligns with the t^{th} frame of V_1 (Note that since V_2 is fully contained inside V_1 , this will always be the case). We first of all manually extract motion regions for the first 50 frames in V_2 and for 50 frames around t in V_1 taking appropriate consideration of the distance of t from the start and the end of the video. A temporal span of 50 frames is imperative since SSMs typically require some decent number of frames to exhibit discriminative patterns. Once the motion regions are extracted in both the videos, part-SSMs (or SSMs) are applied to each of the motion regions in the two videos. If

V_1 contains 2 motion regions M_1 and M_2 , and V_2 contains 3 motion regions N_1 , N_2 and N_3 , then the part-SSM of M_1 is matched against the part-SSMs of all of N_1 , N_2 and N_3 using the recognition procedure of Algorithm 3 and the match with the least distance is accepted. A similar thing is done for M_2 . Once we know which motion regions of V_1 correspond to which regions of V_2 , we synchronize the corresponding motion regions across the two videos and the final indices are averaged to give a potentially refined index t_r . This means that the first frame of V_2 now aligns with the t_r frame of V_1 . Since, we had considered a temporal span of 50 frames for calculating part-SSMs, it might be possible that if the descriptors are erroneous, $t_r > t$. In such a case, the final synchronization index is taken as t only and we report that no refinement has been possible. If $t_r < t$, we get a refined index and the final result is t_r .

Note that the extraction of motion regions for such cases can also be done using some automatic motion segmentation procedures. However, we depict our results using user-assisted motion region extraction in order to remove any scope of error in the final result due to the motion segmentation method. This helps to correctly quantify the refinement procedure deployed.

Algorithm 4 Identify Redundant Self Similarity Structures

1. Initial Setup

- (a) Let S be a Self Similarity Structure (a 2D square image) of size $N \times N$.

2. Take a Frequency Transform

- (a) Take 2D Discrete Fourier Transform (DFT) F of S .
- (b) Take absolute value of F , and normalize this 2D matrix, so that the maximum value is 1. Call this matrix G .

3. Binarize and Threshold

- (a) Convert G into binary numbers based on a threshold μ . For values in G greater than μ , make them 1, else 0.
 - (b) In case, number of unity values in G are greater than ρ percent of the values, render the structure S as redundant, else not.
 - (c) Based on the learning of various Self Similarity structures, $\mu = 0.75, \rho = 0.25$.
-

4. RESULTS AND DISCUSSION

We evaluate our approach on a number of user-generated sports videos of cricket, baseball and football matches. All these videos contain multiple people, do not have very discriminative static background features and the scene is captured from a far-off distance. Examples of videos used for evaluation are given in Fig 7. It can be seen that although some videos (like football) might contain discriminative background features, the same static features are not guaranteed to be available from a different viewing angle, especially when the videos are shot for a small duration of time. This is in contrast to sports like lawn tennis and badminton where the number of people involved are few (two or four), and the scene normally has discriminative static features. In sports like basketball, while the number of people involved is greater, the videos can normally be shot from a close distance in which case estimating poses of the people can be leveraged for synchronization tasks. Our entire dataset contains 20 synchronization tasks (8 from cricket

videos, 6 each from baseball and football videos). Each synchronization task consists of finding the correct temporal alignment for two unsynchronized videos V_1 and V_2 . All videos have a frame rate of 25 fps. The videos have a time duration of approximately 15 seconds on average. Given two unsynchronized videos V_1 and V_2 , we assume V_2 is wholly contained inside V_1 , i.e. searching for the time shift t^* in V_1 that best corresponds to the first frame of V_2 suffices.

Algorithm 5 Form Comparison Matrix between two Self Similarity Structures

1. Initial Setup

- (a) Consider two Self Similarity structures S_1 and S_2 of sizes $f_1 \times f_1$ and $f_2 \times f_2$ respectively.

2. Temporal Localization

- (a) Do temporal localization of S_1 and S_2 around every frame to create $S_{1_{local}}$ and $S_{2_{local}}$. Thus, $S_{1_{local}}$ and $S_{2_{local}}$ will have f_1 and f_2 temporally localized portions respectively.

3. Apply Feature Transform

- (a) For each temporally localized portion in $S_{1_{local}}$ and $S_{2_{local}}$, take HOG-LBP [33]. This will give temporally localized arrays L_1 and L_2 , of sizes $f_1 \times H$ and $f_2 \times H$ respectively, where H is the length of the HOG-LBP descriptor.

4. Form Comparison Matrix

- (a) For the comparison matrix C by taking Euclidean distances between $L_1(i,:); i = 1 \dots f_1$ and $L_2(j,:); j = 1 \dots f_2$. Thus, $C(i,j) = D(L_1(i,:), L_2(j,:)); i = 1 \dots f_1; j = 1 \dots f_2$, where D is the Euclidean distance measure.
-

Table 1 reports the mean synchronization frame error for our test set using the proposed approach (WT), the method of [8] using self-similarity matrices (SSMs), and the algorithm of [18] based on temporal signals from epipolar lines (TE). The TE method of [18] assumes the prior availability of the fundamental matrix, which is not available for the videos in our test set. For evaluating TE, we thus estimate the fundamental matrix as described in Sec 2. It can be seen that our method significantly outperforms the other two methods for all considered classes of sports videos. We have also evaluated our approach by neglecting the sampling of feature points around the edge map (EMS). This is done to show the effectiveness of the EMS approach (Sec 2). It can be seen that without EMS, the WT performance falls-off but still remains better than the state-of-the-art methods. This can be attributed to the fact that although WT without EMS tends to match the plausibly accurate trajectories, it misses out some important discriminative trajectories, e.g. the trajectory of the hand of a person bowling a ball in a cricket/baseball match. Such a trajectory is taken into account when EMS is considered with WT, thereby yielding better results.

The results shown in Table 1 assume that the warping functions estimated were 10th order polynomials ($W = 10$), $\alpha = 0.3$ with Σ having a constant diagonal entry of 0.25 (See Algorithm 1). We now evaluate our approach for varying values of W . The results are shown in Fig 9. It can be seen that the results remain stable for values of W ranging from 8 to 12; however, they deteriorate

	WT + part-SSM	WT	WT + SSM	WT - EMS	TE	SSM
C	12.25	14.25	12.87	18.25	39.25	60.87
B	10.50	13.16	12.25	12.50	27.83	46.33
F	12.16	18.66	17.33	24.33	43.66	73.66

Table 1: (C = Cricket, B = Baseball, F = Football) Synchronization Errors (in frames) using our proposed approach (WT), our proposed approach excluding the random sampling around the edge map (WT - EMS), the TE approach of [18], and the Self-similarity Matrices (SSM) approach of [8]. Our method performs significantly better for these classes of sports videos. The results are best when we consider WT with EMS. This is understandable since EMS enables us to use more discriminative trajectories. Note that the SSM approach here means that the synchronization of the entire video was done with only SSMs. Clearly the results are worst. The results are also there with the refinement procedures applied to our approach WT. It can be seen that the best results are when part-SSMs are used for refining the synchronization indices obtained with WT.

with smaller values of W . This is understandable since the order W accounts for the non-linearity in the warping functions that can relate two given trajectories. Lower values of W are not capable of modelling highly non-linear curves; thus, some trajectories might not be related in an effective manner. Fig 9 also shows the results of our WT approach with varying values of α . A larger α indicates that we have the flexibility to consider a greater number of new feature points while iterating to find the most common set of warping functions possible for a given time shift. A very low value of α thus achieves marginally inferior results since such flexibility is lost, resulting in more erroneous trajectory matches. Fig 9 shows the results with varying values of the variance of the 2D Gaussian function Σ . The variance helps to model the uncertainty of the fundamental matrix. A lower value indicates that the fundamental matrix is being assumed too precise and so few points are considered around the region specified by the fundamental matrix. A larger value of Σ indicates that the fundamental matrix is very uncertain, and thus might lead to more erroneous trajectories being considered for correspondence. However, this has little effect on the results, since most erroneous trajectories are automatically neglected by our approach; hence, the results of Fig 9.

Although our proposed method for temporal synchronization outperforms the state-of-the-art methods by a significant amount, it is not perfect. This can be attributed to the vast dynamic range in user-generated input videos. However, our algorithm still achieves a temporal synchronization with a mean error of around ± 1 seconds and thus can always be used as an initial step to be followed by a refined synchronization procedure.

We now present the results with part-SSMs as described in Section 3. We first of all illustrate that our proposed part-SSMs outperform the state-of-the-art approaches for recognition of cross-view activities under occlusions. We show the results on *IXMAS Occluded* dataset [34]. Snapshots of some sequences from *IXMAS Occluded* dataset are presented in Figure 8. Recognition results on *IXMAS Occluded* dataset are presented in Table 2. The dataset contains all activities from five viewpoints (five cameras - *cam1*, *cam2*, *cam3*, *cam4*, *cam5*) performed by 10 different subjects under occlusions. It can be seen that we outperform the state-of-the-art for most cases. It is noteworthy that while the method of [11] uses training from a single viewpoint, method of [34] uses training

Figure 7: Snapshots of the user-generated sports videos used as our test set. The videos are shot from different viewing angles during cricket, baseball and football matches. In all cases, there are multiple people involved in the scene and the scene is shot from a far-off distance. *From Left to Right (Column Wise)* Cricket, Baseball and Football Videos Note that the snapshots do not show synchronized frames of a scene, they simply illustrate the complexity of videos considered for our evaluation. All the videos in our dataset along with the synchronization tasks formed out of them are available on our website <http://mi.eng.cam.ac.uk/~ss965/>.

data from multiple viewpoints. For evaluating our part-SSM approach, we do training using only a single camera similar to [11]. For all video sequences in these datasets, the frames were resized to 240×160 and spatial tiles of sizes 20×20 were made.

We next apply part-SSMs as a refinement tool on the synchronization results of sports videos as described in Section 3. To draw a comparison between part-SSMs and SSMs for refinement, we also apply SSMs once an approximate measure of synchronization is achieved. The results are shown in Table 1, where it can be observed that least mean error is obtained when the part-SSMs are used with our synchronization procedure. Note that we do not present experimental results of using part-SSMs with other synchronization approaches like that of Temporal Signals (TE), etc. This because part-SSMs as a refinement tool generally do not make a huge correction in the synchronization index, in which case other synchronization approaches can never yield superior results to that of ours.

Figure 8: Snapshots of sequences from *IXMAS Occluded* dataset [34]. The dataset contains single person activities, and some objects in the scenes occlude the activities.

Method	cam1	cam2	cam3	cam4	cam5
Junejo et al. [11]	66.4	67.6	63.6	58.8	56.1
Weinland et al. [34]	76.8	76.4	78.0	78.2	64.7
Part-SSMs (OURS)	74.1	78.2	76.2	81.4	71.2

Table 2: Cross-view activity recognition results with part-SSMs on *IXMAS Occluded* Dataset. Our results have used training data from a single camera only. The column of cam1 indicates that the training data only consisted of the videos shot from the camera viewpoint (1), while the test set contained the videos from all other camera viewpoints (2,3,4,5). A similar explanation goes for other columns. Note that the classification is performed using the nearest-neighbour (NN) measure.

5. CONCLUSIONS AND FUTURE WORK

We have presented a synchronization method that infers the time alignment index between two given user-generated videos of the same sport event. Unlike other approaches, our method does not need a precise knowledge of the fundamental matrices, homographies, poses of people or the feature trajectories. Instead, the proposed method automatically selects the feature points in the two given videos whose 2D trajectories can best be related with a nearly common warping function, and the correct time alignment index is given by the value at which the warping functions are maximally similar. To narrow the search space for finding trajectory correspondences, an approximate estimation of the fundamental matrix suffices. Thus, we are able to model the uncertainties in the feature trajectories and the fundamental matrix along with the inherent 3D motion to find the correct temporal shift. Empirical results have shown that our method significantly outperforms the state-of-the-art methods. We have further proposed a robust view-invariant activity recognition descriptor (part-SSMs) that better characterizes actions under occlusions. Using part-SSMs for refining the synchronization accuracy provided us the best results.

6. REFERENCES

- [1] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.
- [2] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In *ICCV*. IEEE, 2001.
- [3] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *TPAMI*, 24(11):1409–1424, 2002.
- [4] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *IJCV*, 68(1), 2006.
- [5] P. Chen and D. Suter. Simultaneously estimating the fundamental matrix and homographies. *Robotics, IEEE Transactions on*, 25(6):1425–1431, 2009.
- [6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1. IEEE, 2005.
- [7] T. J. Darrell, I. A. Essa, and A. P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(12):1236–1242, 1996.
- [8] E. Dexter and I. Laptev. Multi-view synchronization of human actions and dynamic scenes. In *BMVC*, 2009.
- [9] M. Germann, T. Popa, R. Ziegler, R. Keiser, and M. Gross. Space-time body pose estimation in uncontrolled environments. In *3DIMPVT*. IEEE, 2011.
- [10] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*. IEEE, 2009.

Figure 9: (From Left to Right) Mean Synchronization Error (in frames) using our proposed approach (WT) - With varying values of W (the order of the warping function). The results are almost stable for higher values of W ; but fall off with lower values of W . This can be attributed to the fact that a lower order W is able to model the non-linearity in the warping functions to a lesser extent. With varying values of α . The results remain almost stable and exhibit a decrease in accuracy for very small values of α . This is because a lower value of α limits the number of feature points considered and may therefore miss good trajectory correspondences. With varying values of Σ . The results are almost stable but show a slight decrease in accuracy with lower Σ . This is because a lower value of Σ models the uncertainty in the fundamental matrix to a lesser extent. Best viewed in color.

- [11] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *TPAMI*, 33(1):172–185, 2011.
- [12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*. IEEE, 2011.
- [13] C. Lei and Y.-H. Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *Image Processing, IEEE Transactions on*, 15(9):2473–2480, 2006.
- [14] R. Li, R. Chellappa, and S. K. Zhou. Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. In *CVPR*. IEEE, 2009.
- [15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [16] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. California, USA, 1967.
- [17] K. Mikolajczyk and H. Uemura. Action recognition with appearance–motion features and fast search trees. *Computer Vision and Image Understanding*, 115(3):426–438, 2011.
- [18] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. In *ECCV*. Springer, 2010.
- [19] I. Reid and A. Zisserman. Goal-directed video metrology. In *ECCV*. Springer, 1996.
- [20] P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.
- [21] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*. IEEE, 2012.
- [22] H. J. Seo and P. Milanfar. Action recognition from one example. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):867–882, 2011.
- [23] J. Serrat, F. Diego, F. Lumberras, and J. M. Álvarez. Synchronization of video sequences from free-moving cameras. In *Pattern Recognition and Image Analysis*, pages 620–627. Springer, 2007.
- [24] J. Shi and C. Tomasi. Good features to track. In *CVPR*. IEEE, 1994.
- [25] M. Singh, A. Basu, and M. Mandal. Event dynamics based temporal registration. *Multimedia, IEEE Transactions on*, 9(5):1004–1015, 2007.
- [26] L. Spencer and M. Shah. Temporal synchronization from camera motion. In *ACCV*, 2004.
- [27] C. Sun, I. Junejo, and H. Foroosh. Action recognition using rank-1 approximation of joint self-similarity volume. In *ICCV*. IEEE, 2011.
- [28] J. Tompkin, K. I. Kim, J. Kautz, and C. Theobalt. Videoscapes: exploring sparse, unstructured video collections. *ACM Transactions on Graphics (TOG)*, 31(4):68, 2012.
- [29] K. Tran, I. Kakadiaris, and S. Shah. Part-based motion descriptor image for human action recognition. *Pattern Recognition*, 45(7):2562 – 2572, 2012.
- [30] P. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. In *BMVC*, 2003.
- [31] T. Tuytelaars and L. Van Gool. Synchronizing video sequences. In *CVPR*. IEEE, 2004.
- [32] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*. IEEE, 2011.
- [33] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, pages 32–39. IEEE, 2009.
- [34] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*. Springer, 2010.
- [35] A. Whitehead, R. Laganieri, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In *Workshop on Application of Computer Vision*. IEEE, 2005.
- [36] L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Proc. Workshop on Vision and Modelling of Dynamic Scenes, Copenhagen*, 2002.
- [37] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*. IEEE, 2001.