Feature-Based Sequence-to-Sequence Matching

Yaron Caspi, Denis Simakov and Michal Irani

Dept. of Computer Science and Applied Math

The Weizmann Institute of Science

76100 Rehovot, Israel

Abstract. This paper studies the problem of matching two unsynchronized video sequences of the same dynamic scene, recorded by different stationary uncalibrated video cameras. The matching is done both in *time* and in *space*, where the spatial matching can be modeled by a homography (for 2D scenarios) or by a fundamental matrix (for 3D scenarios). Our approach is based on matching space-time *trajectories* of moving objects, in contrast to matching interest *points* (e.g., corners), as done in regular feature-based image-to-image matching techniques. The sequences are matched in space and time by enforcing consistent matching of all points along corresponding space-time trajectories.

By exploiting the dynamic properties of these space-time trajectories, we obtain sub-frame temporal correspondence (synchronization) between the two video sequences. Furthermore, using trajectories rather than feature-points significantly reduces the combinatorial complexity of the spatial point-matching problem when the search space is large. This benefit allows for matching information across sensors in situations which are extremely difficult when only image-to-image matching is used, including: (a) matching under large scale (zoom) differences, (b) very wide base-line matching, and (c) matching across different sensing modalities (e.g., IR and visible-light cameras). We show examples of recovering homographies and fundamental matrices under such conditions.

Keywords: sequence-to-sequence matching, alignment in space and time, dynamic information, multi-sensor alignment, wide base-line matching, trajectory matching.

1. Introduction

Image-to-image matching methods, e.g., (Faugeras et al., 2001; Hartley and Zisserman, 2000; Xu and Zhang, 1996; Bergen et al., 1992; Szeliski and Shum, 1997; Zhang et al., 1995; Zoghlami et al., 1997), are inherently restricted to the information contained in individual images, i.e., the spatial variations *within* image frames (which capture the scene appearance). But there are cases when there is not enough common spatial information within the two images to allow reliable image matching. One such example is illustrated in Fig. 1. The input images 1.a and 1.b contain a single object, but we want to match (or align) the entire frame. Alignment of image 1.a to image 1.b is not uniquely defined (see Fig. 1.c). However, a video sequence contains much more information than any individual frame does. In particular, a video sequence captures information about scene dynamics such as the trajectory of the moving object shown in Fig. 1.d and 1.e, which in this case provides enough information for unique alignment both in space and in time (see Fig. 1.f). The scene dynamics, exemplified here by trajectories of moving objects, is a property that is inherent to the scene, and is thus common to all sequences recording the same scene, even when taken from

different video cameras. It therefore forms an *additional* or *alternative* powerful cue for matching video sequences.

The benefits of exploiting scene dynamics for matching sequences was noted before. Caspi and Irani (Caspi and Irani, 2000) described a direct (intensity-based) sequence-to-sequence alignment method. Their method is based on finding the space-time transformation which minimizes the intensity differences (SSD) between the two sequences, and was applied to cases where the spatial relation between the sequences could be modeled by a 2D parametric transformation (a homography). It was shown to be useful for addressing rigid as well as complex non-rigid changes in the scene (e.g., flowing water), and changes in illumination. However, that method does not apply when the two sequences have different appearance properties, such as with sensors of different sensing modalities, nor when the spatial transformation between the two sequences is very large, such as in wide base-line matching, or in large differences in zoom.

This paper illustrates a feature-based approach for space-time matching of video sequences. The "features" in our method are space-time trajectories constructed from moving objects. This approach can recover the 3D epipolar geometry between sequences recorded by widely separated video cameras, and can handle significant differences in appearance between the two sequences.

The advantage of our approach over using regular feature-based image-to-image matching is illustrated in Fig. 2. This figure shows two sequences recording several small moving objects. Each feature point in the image-frame of Fig. 2.a (denoted by A-E) can in principle be matched to any other feature point in the image-frame of Fig. 2.b (ignoring matching using local appearance). In this case there is not sufficient information in any individual frame to uniquely resolve the point correspondences. Point trajectories, on the other hand, have additional shape properties which simplify the *trajectory* correspondence problem (i.e., which trajectory corresponds to which trajectory) across the two sequences, as shown in Fig. 2.c and 2.d.

More recently, work has been devoted to development of feature detectors and descriptors which are invariant to severe geometric transformations, such as large changes in scale and rotation (e.g., (Mikolajczyk and Schmid, 2004; Matas et al., 2002; Tuytelaars and Gool, 2004; Ferrari et al., 2003; Kadir et al., 2004; Lowe, 2004; Mindru et al., 2004). This made it possible to find feature correspondences between images even when taken from significantly different view points. However, those methods assume that there is some common spatial information in the vicinity of the feature point. Such an assumption does not hold in extreme cases, such as when the two cameras are opposed to each other. In this work we show that when dynamic information is available, trajectories of moving objects can be used to match features across images,

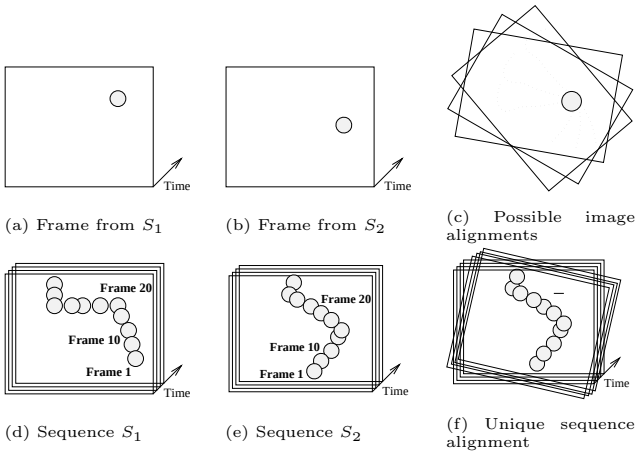
not only in severe cases as handled by previous methods, but also in extreme cases, when no common appearance information is available, such as when the two cameras are facing each other (examples on Figs. 5 and 6).

Stein (Stein, 1998) and Lee et.al. (Lee et al., 2000) described a method for estimating a time shift and a homography between two sequences based on alignment of centroids of moving objects. However, in (Stein, 1998; Lee et al., 2000) the centroids were treated as an *unordered* collection of feature points and not as trajectories. In contrast, we enforce correspondences between *trajectories*, thus avoiding the combinatorial complexity of establishing point matches of all points in all frames, resolving ambiguities in point correspondences, and allowing for temporal correspondences at *sub-frame* accuracy. This is not possible when the points are treated independently (i.e., as a "cloud of points"). Recently, based on the shorter version of our paper (our ECCV'02 Workshop paper - (Caspi et al., 2002)), Stauffer and Tieu (Stauffer and Tieu, 2003) demonstrated how the method of Stein and Lee can indeed be improved using correspondences between *trajectories*.

Section 2 formulates the underlying problem, and Section 3 presents our sequence matching algorithm that is based on matching feature trajectories. The algorithm receives as input two unsynchronized video sequences and simultaneously estimates the parameters of the temporal and spatial transformation (relation) between the two sequences.

Temporal misalignment (unsynchronization) occurs when the two input sequences have a time-shift (offset) between them (e.g., if the cameras were not activated simultaneously), and/or when they have different frame rates (e.g., PAL vs. NTSC). The *spatial* relation between the two sequences results from the camera setups. We have implemented two variants for the two following camera setups: (i) when the spatial relation between the two sequences is a 2D projective transformation (i.e., a homography), and (ii) when the spatial relation between the two sequences is expressed by epipolar geometry (i.e., a fundamental matrix).

Section 4 shows that by replacing point features with trajectories of moving points we can address several cases which are very difficult for regular image-to-image matching. We show that situations that are inherently ambiguous for image-to-image matching methods are often uniquely resolved by the sequence-to-sequence matching approach. In particular, these include situations where there is very little common appearance (spatial) information across the two sequences, such as in sequences of different sensing modalities (e.g., Infra-Red and Visible-light sensors), large scale differences, and wide base-lines between the cameras. We apply our method to such examples, and show that consistency of the scene dynamics (i.e., temporal cues across sequences) can become a major source of information for matching video sequences both in time and in space.



(a) Frame from S_1 (b) Frame from S_2 (c) Possible image alignments

(d) Sequence S_1 (e) Sequence S_2 (f) Unique sequence alignment

Figure 1. **Spatial ambiguities in image-to-image alignment** (a) and (b) show two temporally corresponding frames from two different video sequences viewing the same moving ball. There are infinitely many valid image alignments between the two frames, some of them shown in (c). (d) and (e) display the two sequences of the moving ball. There is only one valid alignment of the two trajectories of the ball. This uniquely defines the alignment both in time and in space between the two video sequences (f).

2. Problem Formulation

Let S and S' be two input image sequences, where S denotes the "reference" sequence, and S' denotes the second sequence. Let $\vec{\mathbf{x}} = (x, y, t)$ be a *space-time point* in the reference sequence S (namely, a pixel (x, y) at frame (time) t) and let $\vec{\mathbf{x}}' = (x', y', t')$ be the matching space-time point in sequence S'. The recorded scene can change dynamically, i.e., it can include moving objects. The cameras can be either stationary or jointly moving with fixed (but *unknown*) internal and relative external parameters. In this setup correspondences in *time* and in *space* between the video sequences can be described/modeled by a small set of parameters $\vec{P} = (\vec{P}_{spatial}, \vec{P}_{temporal})$. Our goal is to recover these parameters.

The specific models that we address and their parameters are discussed next.

Temporal misalignment results when the two input sequences have a time-shift (offset) between them (e.g., if the cameras were not activated simultaneously), and/or when they have different frame rates (e.g., PAL vs. NTSC). Such temporal misalignments can be modeled by a 1-D affine transformation in time $t' = s \cdot t + \Delta t$, and is typically at *sub-frame* time units. Note that in most cases s is known – it is the ratio between the frame rates of the two cameras (e.g., for PAL and NTSC sequences, it is $s = 25/30 = 5/6$). Therefore, in such cases $\vec{P}_{temporal}$ contains only one unknown parameter, Δt.

To model the spatial parameters let us look at the spatial part of a space-time point. Let $\vec{p}(t) = (x, y, 1)^T$ denote the homogeneous coordinates of only the *spatial* component of a space-time point $\vec{\mathbf{x}} = (x, y, t)$ in S. The *spatial misalignment* between two sequences results from the fact that the two cameras have different external and internal calibration parameters. We will consider two possible cases: the *2D case* and the the *3D case*:

(i) By the *2D case* we refer to the case where the distance between the camera projection centers is negligible relative to the distances of the cameras to the scene, or else if the scene is roughly planar. In this *2D case* the space-time relation between the two sequences is expressed by

an unknown 3×3 homography H and the unknown Δt:

$$H\vec{p}(t) \cong \vec{p}\,'(s \cdot t + \Delta t).$$

In this case the nine spatial parameters

$$\vec{P}_{spatial} = [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ h_{31}\ h_{32}\ h_{33}]$$

are defined up to a scale factor (h_{ij} are the 9 entries of H)[1], and $\vec{P}_{temporal} = \Delta t$.

(ii) By the *3D case* we refer to the case where the cameras are disjoint and the scene contains observable 3D variations. In this case the space-time relation between the two sequences is expressed by an unknown fundamental matrix F and the unknown Δt :

$$\vec{p}\,'(s \cdot t + \Delta t)^T F \vec{p} = 0,$$

where $[\cdot]^T$ denotes the transpose of a vector. In this case the spatial relation parameters are: $\vec{P}_{spatial} = [f_{11}\ f_{12}\ f_{13}\ f_{21}\ f_{22}\ f_{23}\ f_{31}\ f_{32}\ f_{33}]$, where f_{ij} are the 9 entries of the 3×3 fundamental matrix F (up to a scale factor), and $\vec{P}_{temporal} = \Delta t$.

Note that in either case, F or H are shared by all temporally corresponding pairs of frames because the cameras are fixed relative to each other (both internal parameters and inter-camera external parameters are fixed).

[1] The modification to other 2D parametric models, such as translation, similarity or affine, is trivial (e.g., set $h_{31} = h_{32} = 0$ for a 2D affine model).

(a)Frame from S_1 (b) Frame from S_2 (c) Sequence S_1 (d) Sequence S_2

Figure 2. **Point correspondences vs. trajectory correspondences.** (a) and (b) display two frames out of two sequences recording five small moving objects (marked by A,B,C,D,E). (c) and (d) display the trajectories of these moving objects over time. When analyzing only single frames, it is difficult to determine the correct point correspondences across images. However, point trajectories have additional properties, which simplify the correspondence problem across two sequences (both in space and in time).

3. The Trajectory-Based Sequence Matching Algorithm

Feature-based image matching can be generalized to feature-based sequence matching by extending the notion of features from *feature points* to *feature trajectories*. Let $\gamma = \{\vec{x}_{t_0}, \vec{x}_{t_1}, \ldots, \vec{x}_{t_n}\}$ be a space-time trajectory (remember that by $\vec{x} = (x, y, t)$ we denote space-time points). Denote by Γ and Γ' the sets of all trajectories in sequences S and S' respectively, then spatio-temporal matching between the two sequences can be recovered by establishing correspondences between trajectories from the sets Γ and Γ'.

In particular, a single pair of (non-trivial) corresponding trajectories[2] γ and γ' can uniquely define: (i) the spatial relation, (ii) the temporal relation, (iii) can provide a convenient residual error measure: $err(\vec{P}) = \sum_{\vec{x}\in\gamma} D(\vec{x}, \vec{x}\,') = \sum_{t\in[t_0,\ldots,t_n]} d(\vec{p}(t), \vec{p}\,'(t'))$, where $[t_0, \ldots, t_n]$ is the temporal support of the space-time trajectory γ, $\vec{p}(t)$ is the spatial

[2] By a non-trivial trajectory we mean that it covers a large enough image region, and that its points do not all belong to a degenerate configuration (e.g, a straight line for a homography, or a plane for a fundamental matrix).

position (i.e., pixel coordinates) of the space-time point $\vec{x}$ at time t (in homogeneous coordinates), and $\vec{p}\,'(t')$ is the spatial position of $\vec{x}\,'$ in the other sequence at time $t' = s \cdot t + \Delta t$.

For the homography (2D) case the error measure is: $d(\vec{p}, \vec{p}\,') = dist_H(H\vec{p}(t), \vec{p}\,'(s \cdot t + \Delta t))$, where $dist_H(q_1, q_2)$ is the distance between two points after normalizing each by its third coordinate. For the fundamental matrix (3D) case the error measure is: $d(\vec{p}, \vec{p}\,') = dist_F(F\vec{p}(t), \vec{p}\,'(s \cdot t + \Delta t))$, where $dist_F(l, q)$ is the distance (in pixels) between a point q and a line l (an epipolar line).

We next outline the feature-based sequence-to-sequence alignment algorithm that we have used in our experiments (which is a RANSAC/MDS based algorithm - see (Fischler and Bolles, 1981; Hampel et al., 1986)). Each step of the algorithm is then explained in more detail below:

(1) Construct feature trajectories (i.e., detect and track feature points for each sequence).

(2) For each trajectory estimate its basic properties (e.g., dynamic vs. static, see more examples below).

(3) Based on basic properties construct an initial table of tentative matching between trajectories.

(4) Estimate candidate parameter vector $\vec{P} = (P_{spatial}, P_{temporal})$ by repeatedly choosing (at random) a pair of possibly corresponding trajectories[3]. At each trial compute the set of parameters $\vec{P}$ which mini-

[3] If these are roughly along a straight line choose an additional pair.

mizes the error function $err(\vec{P})$ defined above.

(5) Assign a score for each candidate set of parameters $\vec{P}$ to be the number of corresponding pairs of trajectories whose residual error (or median residual error) is small.

(6) Repeat steps (4) and (5) N times.

(7) Choose $\vec{P}$ which has the highest score.

(8) Refine $\vec{P}$ using all trajectory pairs that supported this candidate.

In our current implementation trajectories of moving objects were computed (Step 1) by tracking unique points on blobs of moving objects. This was done either by tracking the center of mass of moving objects, or the top point on the silhouettes of moving objects[4]. The reliability of the center of mass to be used as a feature point is discussed in (Lee et al., 2000), and the reliability of extreme points on silhouettes is discussed in (Wong and Cipolla, 2001). The KLT feature tracker (Lucas and Kanade, 1981; Tomasi and Kanade, 1991) may also be used to generate additional feature trajectories. In the presence of many trajectories, trajectory properties may be used to reduce the matching complexity (Step 2). For example, dynamic trajectories (of moving objects) in one sequence are matched only against dynamic trajectories in the other sequence. When the cameras are expected to have similar photometric properties, the spatial properties of the features may also be used (e.g., the size or color distribution of the moving object). When

[4] Implicitly assuming that the cameras are horizontal, and the object tip is not occluded in one camera.

we anticipate a significant change in appearance, shape properties of the *trajectories* could still be used (e.g., normalized length, average speed, curvature, 5-point projective invariance (Mundy and Zisserman, 1992)). Although some of these are not projective invariants, they are useful in an initial search for crude tentative matching (Step 3).

A matching of a *single* pair of trajectories across the two sequences induces *multiple* point correspondences across the camera views. These point correspondences are used for computing the spatial and temporal relation between the two sequences. To evaluate a candidate parameter vector $\vec{P} = (h_{11}, \cdots, h_{33}, \Delta t)$, or $\vec{P} = (f_{11}, \cdots, f_{33}, \Delta t)$ (where $h_{11}, \cdots, h_{33}$ or $f_{11}, \cdots, f_{33}$ are the components of a homography H, or a fundamental matrix F, respectively), we minimize the following error function (Step 4 and Step 8) :

$$\vec{P} = \mathop{\mathbf{argmin}}_{\vec{P}} \sum_{\gamma \in \Gamma} \sum_{t \in support(\gamma)} d(\vec{p}(t), \vec{p}\,'(s \cdot t + \Delta t)) \qquad (1)$$

where $d(\cdot)$ is either $dist_H(\cdot)$ or $dist_F(\cdot)$, depending on whether the scene is 2D or 3D (in Step 4 the summation is only over the selected trajectory). The minimization of Eq.(1) is performed by iterating the following two steps:

(i) Fix Δt and approximate H (or F) using standard methods (e.g., (Hartley and Zisserman, 2000) Chapters 3 and 10, respectively).

(ii) Fix H (or F) and refine Δt. Since $t' = s \cdot t + \Delta t$ is not necessarily an integer value (allowing a sub-frame time shift), it is interpolated from

the adjacent (integer time) point locations: $t_1 = \lfloor t' \rfloor$ and $t_2 = \lceil t' \rceil$. We search for $\alpha = t' - t_1$ $(1 \geq \alpha \geq 0)$ that minimizes the following term:

$$\sum_{\gamma \in \Gamma} \sum_{t \in support(\gamma)} d\Big(\vec{p}(t)\ ,\ \vec{p}\,'(t_1) \cdot (1-\alpha) + \vec{p}\,'(t_2) \cdot \alpha\Big) \qquad (2)$$

In our implementation we used a bounded number of refinement iterations (10 to 20), or stopped earlier if the residual error did not change. An initial (integer) approximation for Δt was derived using exhaustive search over a small fixed temporal interval (20-25 frames in our experiments).

Examples of applying the above algorithm to video sequences of different scenarios are found in Figs. 3,4,5,6, (see figure captions for further details).

4. Benefits of Feature-Based Sequence Matching

When there are *no* dynamic changes in the scene, sequence-to-sequence matching provides no benefit over image-to-image matching. The increase in the data size (sequences vs. images) only increases the signal-to-noise ratio, but does not provide new information. On the contrary, some degenerate cases may result in space-time ambiguities, see (Caspi and Irani, 2000; Giese and Poggio, 2000; Stein, 1998). However, when the scene dynamics is rich enough to exclude such ambiguities (see Section 3.2 in (Caspi and Irani, 2000)), sequence matching is superior

to image matching in multiple ways. Below we mention some of its benefits:

(i) Resolving Spatial Ambiguities. Inherent ambiguities in image-to-image matching occur, for example, when there is insufficient common appearance information across images. This can occur when there is not enough spatial information in the scene, such as in the case of the small ball against a uniform background in Fig. 1. Limited common appearance information across images can also occur when the two cameras record the scene at significantly different zooms (such as in Fig. 4.a and 4.b), thus observing different features at different scales. It can also occur when the two cameras have different sensing modalities (such as the Infra-Red and visible-light cameras in Fig. 3.a and 3.b), thus sensing different features in the scene. A number of approaches have been proposed for such cases (e.g., (Lowe, 2004; Mindru et al., 2004)). Yet, they still rely on some appearance similarity. However, there are extreme cases, such as when the two cameras face each other, when there is very little or *no* common appearance information.

In contrast, trajectories of moving objects over time are independent of the sensor or appearance properties and therefore form a powerful cue for matching across the two sequences, even in extreme cases such as the moving people or the ball in Fig. 6. The need for consistent appearance information is replaced by consistent temporal behavior,

as captured by trajectories of moving objects estimated *within* each sequence separately.

(ii) Improved Accuracy for Unsynchronized Video. Even when there is sufficient spatial information within the images, and accurate frame correspondences is known between the two sequences, sequence-to-sequence matching may still provide higher accuracy in the estimation of the spatial transformation than image-to-image matching. This is true even when all the spatial constraints from all pairs of corresponding images across the two sequences are simultaneously used to solve for the spatial transformation. This is because image-to-image matching is restricted to matching of existing physical frames, whereas these may not have been recorded at exactly the same time due to *sub-frame* temporal misalignment between the two sequences. Sequence-to-sequence matching, on the other hand, is not restricted to physical ("integer") image frames. It can thus *spatially* match information across the two sequences at sub-frame temporal accuracy. This leads to higher sub-pixel accuracy in the spatial matching/alignment.

This phenomenon is mostly noticeable when the scene is highly dynamic. Fig. 7 shows such an example. Importance of recovery of sub-frame time differences for accuracy improvement has also been recently reported in (Tresadern and Reid, 2003), in the context of capturing human motion.

(iii) Reduced Combinatorial Complexity. The combinatorial complexity of a matching algorithm depends on the following factors:

(a) the probability of detecting the same features in both images ("re-detection"),

(b) the probability of finding correct feature matches across the two images ("unique descriptors"),

(c) the minimal number of feature matches that are required for computing the transformation (homography or fundamental matrix).

Recent methods propose sophisticated detectors and descriptors (e.g., (Mikolajczyk and Schmid, 2004; Lowe, 2004)), which decrease the complexity of wide-baseline matching by increasing the probabilities of (a) and (b) above. Nevertheless, in extreme wide-baseline cases, such as when the two cameras are facing each other, the probability of correct feature matching across the two images will still be very low (the probabilities of both (a) and (b) will be low in this case). However, when using trajectories as features, the probability of (a) and (b) remain high.

Moreover, the main difference in complexity between sequences-to-sequence matching and image-to-image matching results from the difference in the minimal number of correspondences that are required for computing the transformations, i.e., from item (c) above.

Let m be the minimal number of correspondences required for computing a spatial transformation $\vec{P}_{spatial}$. For a homography $m = 4$ and for a fundamental matrix $m = 7$ (or $m = 8$ if linear 8-point algorithm is used). Let ϵ be the probability that a feature matching across the two images is correct (and therefore the probability of a mismatch or an outlier is $(1 - \epsilon)$). ϵ results from both (a) and (b) above. A RANSAC-like matching algorithm requires that at least one of the trials (i.e., one random sampling of m correspondences) will not contain any mismatches (outliers). Let N be the number of trials that are required to ensure with probability p (usually $p = 99\%$) that at least one random sample of m features is free from mismatches. Then N is given by the following formula (Rousseeuw, 1987; Hartley and Zisserman, 2000):

$$N \geq \frac{log(1-p)}{log(1-\epsilon^m)}. \tag{3}$$

This formula emphasizes why in a standard image-to-image matching accurate candidate feature correspondences are crucial (e.g., SIFT feature descriptor uses 128 values for each point to increase this probability).

If, however, ϵ is small (such as in Fig. 5), then having a small m is crucial for keeping the complexity low. For example, assume $\epsilon = \frac{1}{100}$, then according to Eq. (3) the number of necessary trials for computing a homography ($m = 4, \epsilon = \frac{1}{100}, p = 99\%$) is $N \geq 4.6 \times 10^8$. On the other hand, in the case of sequence-to-sequence matching, one pair of

corresponding trajectories is enough[5], therefore $m = 1$, and therefore according to Eq. 3 the number of trials is reasonable: $N \geq 459$.

When dealing with *unsynchronized* video sequences we should also take into account the temporal ambiguity. Thus, for each pair of corresponding trajectories, we further have to verify T possible matches, where T is the range of possible temporal misalignments. Therefore, the number of trials for an unsynchronized pair of sequences is $O(T \cdot N)$ (in our experiments we usually allow for $T = 25$ frames, i.e., we assume that the temporal offset between the two sequences is at most $\Delta t = 1$ second).

When only trajectories of moving objects are used, the number of trajectories is usually very small, leading to an additional reduction in the complexity of trajectory matching (by increasing ϵ). Furthermore, when moving objects appear at different times in the sequence, the complexity of trajectory matching is even further reduced.

5. Applications and Results

The above mentioned benefits of sequence-to-sequence matching/alignment give rise to new video applications, that are very difficult or even impossible to obtain using existing image-to-image matching tools. Some of these are illustrated in figures 3, 4, 5, 6 and briefly outlined below.

[5] A simple trajectory match induces many point matches (of all the points on the trajectory)

For viewing the complete video sequences, see:

http://www.wisdom.weizmann.ac.il/∼vision/traj2traj.html

(i) Multi-sensor alignment. The same objects look different in visible and infra-red light, which often makes impossible to match them across the views relying on their appearance. For example, 200 features were extracted in the multi-sensor image pair of Fig. 3.a and Fig. 3.b using Harris corner detector (Harris and Stephens, 1988), but only two out of the 200 turned out belonging to the same real world point. On the other hand, if we detect and track a moving object in both views, then its trajectory no longer depends on the sensing modality of the camera, and thus forms a powerful dynamic cue for alignment. An example of multi-sensor sequence-to-sequence alignment is presented in Fig. 3. (In this case a homography was computed).

(ii) Matching across significant zoom differences. Fig. 4 shows an example of aligning sequences obtained at significantly different zooms. Due to the scale difference (1 : 3) the search range for corresponding features is large (the same features appear at distant locations in the images). Furthermore, the scene is captured at significantly different spatial resolutions and lacks prominent spatial structure in the overlapping region of the two images, which makes the matching of conventional features problematic. The homography was accurately

recovered using sequence-to-sequence alignment. See caption of Fig. 4 for more details.

(iii) Wide base-line matching. Another difficult scenario for image matching is the wide base-line case. When cameras capture the scene from distant viewpoints, they see objects from different sides. We took extreme examples of two cameras, situated on the opposite sides of the scene (i.e., the cameras are facing each other; in fact each camera sees the other camera). The cameras observe the same objects, but can never see the same point.

Our algorithm succeeds to recover the fundamental matrix in this situation with reasonable accuracy, as shown in Figs. 5 and 6 (see figure captions for more information).

(iv) New video applications. Unsynchronized video sequences can be temporally matched (synchronized) at *sub-frame* accuracy. Such sub-frame synchronization gives rise to new video applications including super-resolution *in time* (Shechtman et al., 2002), where multiple video sequences with low temporal resolution (low frame-rate) are combined into a single high temporal resolution (high frame-rate) output sequences.

Acknowledgments

This work was partially supported by the European Commission (VIBES Project IST-2000-26001).

References

Bergen, J., P. Anandan, K. Hanna, and R. Hingorani: 1992, 'Hierarchical Model-Based Motion Estimation'. In: *European Conference on Computer Vision (ECCV)*. Santa Margarita Ligure, pp. 237–252.

Burt, P. and R. Kolczynski: 1993, 'Enhanced image capture through fusion'. In: *International Conference on Computer Vision (ICCV)*. Berlin, pp. 173–182.

Caspi, Y. and M. Irani: 2000, 'A Step Towards Sequence-to-Sequence Alignment'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hilton Head Island, South Carolina, pp. 682–689.

Caspi, Y., D. Simakov, and M. Irani: 2002, 'Feature-Based Sequence-to-Sequence Matching'. In: *ECCV, VAMODS Workshop*. Copenhagen.

Faugeras, O., Q. Luong, and T. Papadopoulo: 2001, *The Geometry of Multiple Images*. MIT Press.

Ferrari, V., T. Tuytelaars, and L. V. Gool: 2003, 'Wide-baseline multiple-view Correspondences'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. Madison, Wisconsin, pp. 718–725.

Fischler, M. A. and R. Bolles: 1981, 'RANSAC Random Sample Concensus: a paradigm for model fitting with applications to image analysis and automated cartography'. In: *Communications of the ACM*, Vol. 24. pp. 381–395.

Giese, M. A. and T. Poggio: 2000, 'Morphable models for the analysis and synthesis of complex motion patterns'. *International Journal of Computer Vision* **38**(1), 59–73.

Hampel, F., P. Rousseeuw, E. Ronchetti, and W. Stahel: 1986, *Robust Statistics: The Approach Based on Influence Functions*. New York: John Wiley.

Harris, C. and M. Stephens: 1988, 'A combined corner and edge detector'. In: *4th Alvey Vision Conference*. pp. 147–151.

Hartley, R. and A. Zisserman: 2000, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge university press.

Kadir, T., A. Zisserman, and M. Brady: 2004, 'An affine invariant salient region detector'. In: *European Conference on Computer Vision (ECCV)*. Prague, Czech Republic.

Lee, L., R. Romano, and G. Stein: 2000, 'Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame'. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* **22**(Special Issue on Video Surveillance and Monitoring), 758–767.

Lowe, D. G.: 2004, 'Distinctive Image Features from Scale-Invariant Keypoints'. *International Journal of Computer Vision* **60**(2), 91–110.

Lucas, B. and T. Kanade: 1981, 'An iterative image registration technique with an application to stereo vision'. In: *Image Understanding Workshop*. pp. 121–130.

Matas, J., O. Chum, U. Martin, and T. Pajdla: 2002, 'Robust wide baseline stereo from maximally stable extremal regions'. In: P. L. Rosin and D. Marshall (eds.): *British Machine Vision Conference*, Vol. 1. London, UK, pp. 384–393.

Mikolajczyk, K. and C. Schmid: 2004, 'Scale and affine invariant interest point detectors'. *International Journal of Computer Vision* **60**(1), 63–86.

Mindru, F., T. Tuytelaars, L. V. Gool, and T. Moons: 2004, 'Moment invariants for recognition under changing viewpoint and illumination'. *Comput. Vis. Image Underst.* **94**(1-3), 3–27.

Mundy, J. and A. Zisserman: 1992, 'Geometric Invariance in Computer Vision'. In: *MIT Press*.

Rousseeuw, P.: 1987, *Robust Regression and Outlier Detection*. New York: Wiley.

Shechtman, E., Y. Caspi, and M. Irani: 2002, 'Increasing video resolution in time and space'. In: *European Conference on Computer Vision (ECCV)*. Copenhagen.

Stauffer, C. and K. Tieu: 2003, 'Automated Multi-Camera Planar Tracking Correspondence Modeling'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Madison, Wisconsin, pp. 259–266.

Stein, G. P.: 1998, 'Tracking from Multiple View Points: Self-calibration of Space and Time'. In: *DARPA IU Workshop*. Montery CA, pp. 1037–1042.

Szeliski, R. and H.-Y. Shum: 1997, 'Creating full view panoramic image mosaics and environment maps'. In: *Computer Graphics Proceedings, Annual Conference Series*. pp. 251–258.

Tomasi, C. and T. Kanade: 1991, 'Detection and Tracking of Point Features'. Technical Report CMU-CS-91-132, Carnegie Mellon University.

Tresadern, P. and I. Reid: 2003, 'Synchronizing Image Sequences of Non-Rigid Objects'. In: *British Machine Vision Conference*, Vol. 2. Norwich, pp. 629–638.

Tuytelaars, T. and L. V. Gool: 2004, 'Matching Widely Separated Views Based on Affine Invariant Regions'. *International Journal of Computer Vision* **59**(1), 61–85.

Wong, K.-Y. K. and R. Cipolla: 2001, 'Structure and Motion from Silhouettes'. In: *International Conference on Computer Vision (ICCV)*, Vol. II. Vancouver, Canada, pp. 217–222.

Xu, C. and Z. Zhang: 1996, *Epipolar Geometry in Stereo, Motion and Object Recognition*. Dordecht, The Netherlands: Kluwer Academic Publishers.

Zhang, Z., R. Deriche, O. Faugeras, and Q. Luong: 1995, 'A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry'. *Artificial Intelligence* **78**, 87–119.

Zoghlami, I., O. Faugeras, and R. Deriche: 1997, 'Using geometric corners to build a 2d mosaic from a set of images'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 420–425.

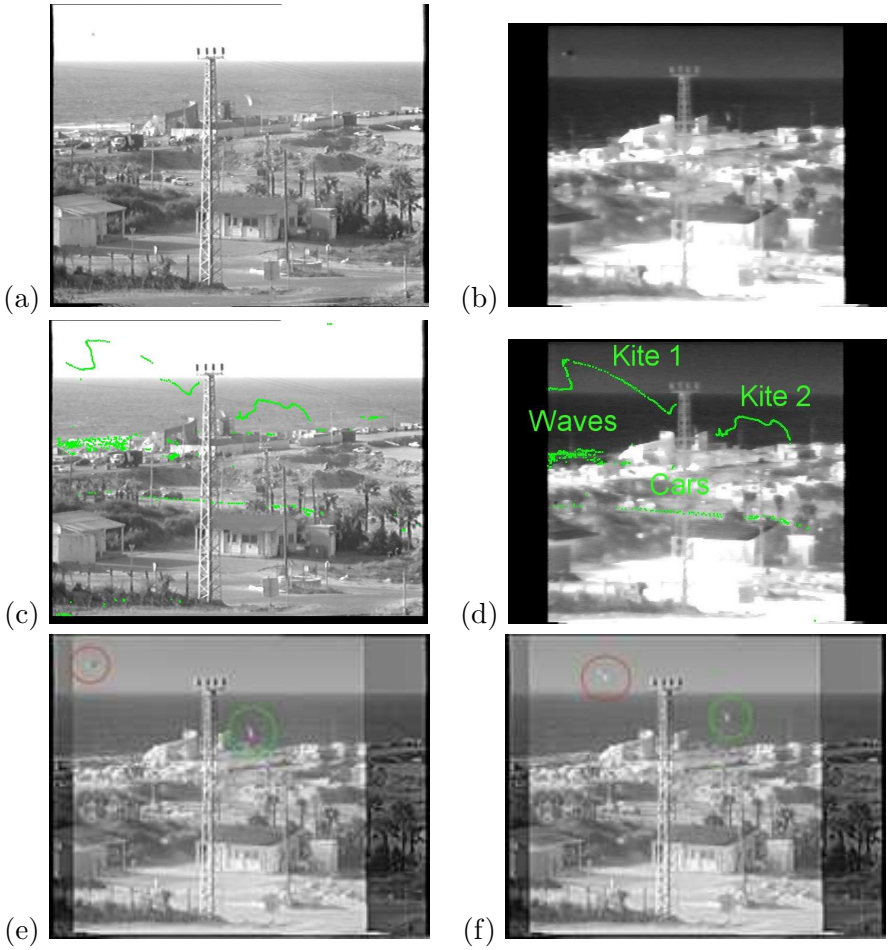
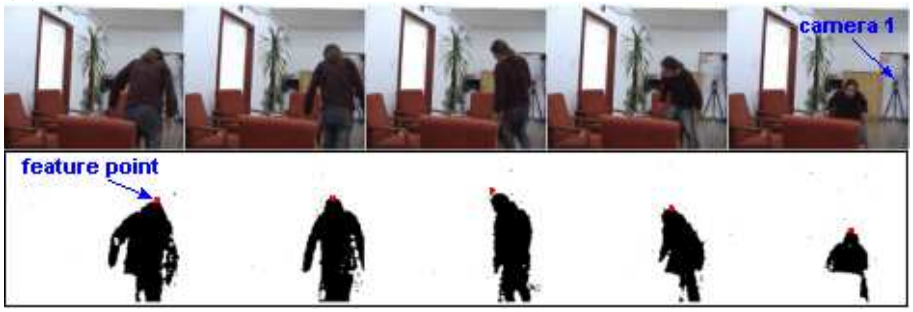


Figure 3. **Multi-Sensor Alignment.** *(a) and (b) display representative frames from a PAL visible light sequence and an NTSC Infra-Red sequence, respectively. The scene contains several moving objects: 2 kites, 2 moving cars, and sea waves. The trajectories induced by tracking the moving objects are displayed in (c) and (d). The two camera centers were close to each other, therefore the spatial transformation was modeled by a homography. The output after spatio-temporal alignment via trajectories matching (Section 3) is displayed in (e) and (f). The recovered temporal misalignment was 1.31 sec. The results are displayed after fusing the two input sequences (using Burt's fusion algorithm (*Burt and Kolczynski, 1993*)). We can now observe spatial features from both sequences. In particular note the right kite which is more clearly visible in the visible-light sequence (circled in light/green), and the left kite which is more clearly visible in the IR sequence (circled in dark/red).*

Figure 4. **Alignment of sequences obtained at different zooms.** *Columns (a) and (b) display four representative frames from the reference sequence and second sequence, showing a ball thrown from side to side. The sequence in column (a) was captured by a wide field-of-view camera, while the sequence in column (b) was captured by a narrow field-of-view camera. The cameras where located next to each other (the spatial transformation was modeled by a homography) and the ratio in zooms was approximately* $1 : 3$*. The two sequences capture features at significantly different spatial resolutions, which makes the problem of inter-camera image-to-image alignment very difficult. The dynamic information (the trajectory of the ball's center of gravity), on the other hand, forms a powerful cue for alignment both in time and in space. Column (c) displays superposition of corresponding frames* after *spatio-temporal alignment, using the algorithm of Section 3 for estimating the homography and the temporal correspondence between the two sequences. The dark (pink) boundaries in (c) correspond to scene regions observed only by the reference (zoomed-out) camera.*

(a) **First camera sequence**:

(b) **Second camera sequence**:



(c)

(d)

Figure 5. **Wide Base-Line Matching** *Rows (a) and (b) display a few corresponding frames of one person (out of three that took part in the experiment) walking and sitting in a hall. The sequences were taken from two opposite sides of the hall. Each camera is visible by the other camera and is marked on the right-most frame by an arrow. Using background subtraction we extract moving objects (people), and select their head tips (the highest point on the silhouette) as feature points (this is illustrated for the second sequence in (b)). The recovered epipolar geometry is displayed in (c) and (d). Static points and their epipolar lines are displayed for verification only and were not used in the computation. Note that the recovered epipoles (the intersection of the epipolar lines) fall very close to their true locations (which is the position of the other camera, marked by a white cross). In this example only one person at a time enters the scene, thus the trajectory correspondence problem becomes simple. An initial temporal alignment with accuracy within one second (25 frames) was manually provided, and the final recovered temporal shift was -2.8 frames. Input data consisted of 7 trajectories with the total of 990 points (from the* 1^{st} *camera) and 8 trajectories with 1322 points (from the* 2^{nd} *camera). Average distance to the recovered epipolar lines was about 0.01 pixels.*

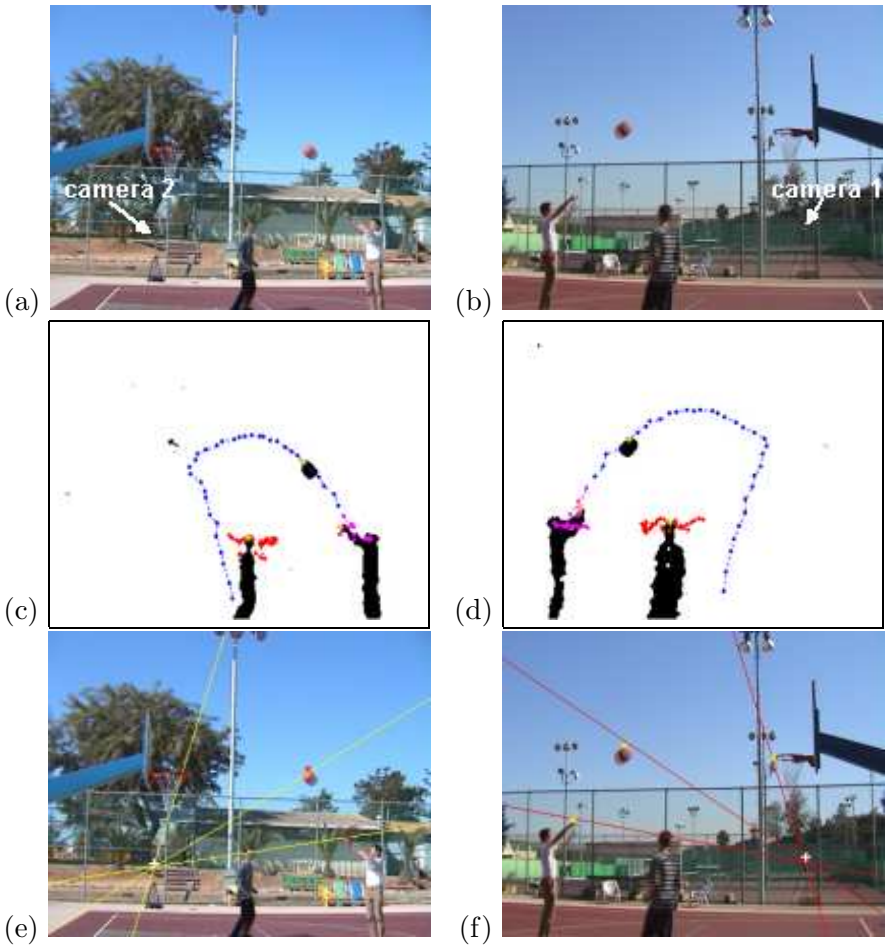


Figure 6. **Wide Base-Line Matching** *(a) and (b) display two representative frames from two sequences of a basketball game taken from two opposite sides of the basket field (the cameras are facing each other). Each camera is visible by the other camera and is circled and marked by a white arrow. Space-time trajectories induced by moving objects (ball and two players) are displayed in (c)-(d) (in different colors for the different objects). The recovered epipolar geometry is displayed in (e) and (f). Points and their epipolar lines are displayed in each image for verification. Note, that the only static objects that are visible in both views are the basket ring and the board. Accuracy of the recovered spatial alignment can be appreciated by the closeness of each point to the epipolar line of its corresponding point, as well as by comparing the intersection of epipolar lines with the ground truth epipole marked by a cross (which is the other camera). In this example the relative blob size of the moving objects was used to provide initial correspondence between the trajectories across the two sequences. Two trajectories (instead of one) were used on each RANSAC iteration, as most trajectories are planar. An initial temporal alignment with accuracy within one second (25 frames) was manually provided, and the final recovered temporal shift was 3.7 frames.*
Input data consisted of 47 trajectories with the total of 2613 points (from the 1^{st} camera) and 47 trajectories with 2582 points (from the 2^{nd} camera). Average distance to the recovered epipolar lines was about 0.01 pixels.

(a) In sequence 1:

(b) In sequence 2:

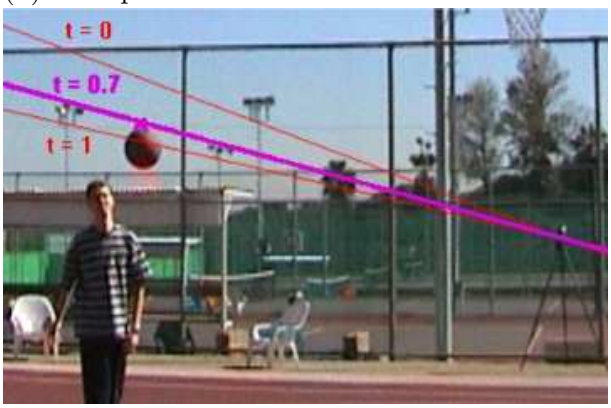


Figure 7. **Subframe temporal synchronization (a)** *displays superposition of the moving ball position in two consecutive frames in sequence 1 (at* $t = 0$ *and* $t = 1$*). The ball is falling at a high speed, thus its displacement is quite noticeable. The feature point is the tip point of the ball in each frame (the highest point on the ball). The dashed blue circle displays the interpolated ball location at the correct time shift (i.e., the correct sub-frame time unit at which the corresponding frame was recorded in the other sequence – sequence 2). In this example it is 0.7 of a frame time, since the global temporal matching was 3.7 frames offset.* **(b)** *The light/red lines display the epipolar lines generated on the image plane of sequence 2 by the "physical" ball in sequence 1 (imaged at "integer" frames* $t = 0$ *and* $t = 1$*). The dark/magenta line displays the epipolar line corresponding to the interpolated location of the ball at* $t = 0.7$*. It can be clearly seen that the ball's feature point (its tip) in sequence 2 is on the epipolar line corresponding to the virtual point (its location at* $t = 0.7$ *in sequence 1). This example also shows that a large error can be introduced by matching only "integer" frames across two sequences (while ignoring the sub-frame temporal offset).*