

Feature-Based Sequence-to-Sequence Matching

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 2D homography or a (3D) fundamental matrix. 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 to match 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.

1 Introduction

Image-to-image matching methods (e.g., [4, 9, 19, 1, 16, 20, 21]) 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, represented as 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 [3] described a direct-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 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, 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 describes a feature-based approach to space-time matching of video sequences which addresses those cases. 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 differences in appearance between the two sequences.

The advantage of this 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. There is no 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.

Stein [15] and Lee et.al. [10] described a method for estimating a time shift and a homography between two sequences based on alignment of centroids of moving objects. Moving objects were detected and tracked in each sequence and their centroids computed. However, there is a fundamental difference between [15, 10] and our approach. The centroids in [15, 10] were treated as an *unordered* collection of feature points and not as trajectories. The spatio-temporal transformation between the two sequences was accordingly computed by examining all possible pairings of corresponding centroids within a time interval. 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”).

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 and NTSC). The *spatial* relation between the two sequences results from the camera setups. We have implemented two variants for 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.

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

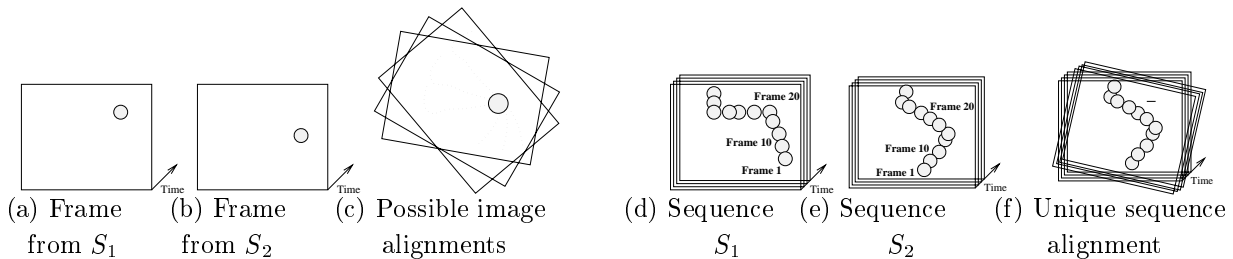


Figure 1: **Spatial ambiguities in image-to-image alignment** (a) and (b) show two temporally corresponding frames from two different video sequences viewing the same moving ball. There are infinitely many valid image alignments between the two frames, some of them shown in (c). (d) and (e) display the 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).

by a small set of parameters $\vec{P} = (\vec{P}_{spatial}, \vec{P}_{temporal})$. Our goal is to recover these parameters. These models 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 and NTSC). Such temporal misalignments can be modeled by a 1-D affine transformation in time $t' = s \cdot t + \Delta t$, and is 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, $\vec{P}_{temporal}$ 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 $\vec{x} = (x, y, t)$ 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 *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 from the scene, or else if the scene is roughly planar. In this case 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 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 $\vec{P}_{temporal} = \Delta t$.

¹The 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).

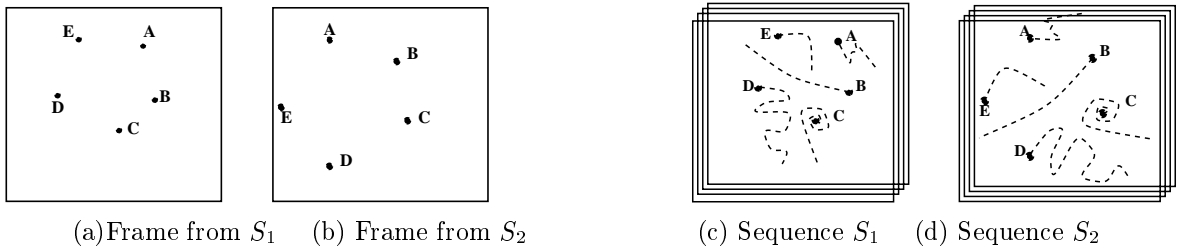


Figure 2: **Point correspondences vs. trajectory correspondences.** (a) and (b) display two frames out of two sequences recording five small moving objects (marked by A,B,C,D,E). (c) and (d) display the trajectories of these moving objects over time. When analyzing only single frames, it is difficult 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).

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).

3 A Feature-Based Sequence Matching Algorithm

Typical feature-based *image* matching methods (e.g., [4, 9, 19]) first apply a local operator to detect interest points in a pair of images (e.g., the Harris corner detector [8]). Once interest points are extracted in the two images, robust estimation methods, such as RANSAC [5], LMS [7], etc., are used for finding corresponding points and recovering the spatial relation between the two images. In some cases [19] a correlation based matching is used to initialize the approximation of matching features. In general, correlation may be based on any property of a feature point, but usually it is based on brightness values of small neighborhoods of the feature points.

Feature-based image matching can be generalized to feature-based sequence matching by extending the notion of features from *feature points* into *feature trajectories*. Let $\gamma(t) = \{\vec{x}_{t_0}, \vec{x}_{t_1}, \dots, \vec{x}_{t_n}\}$ be a space-time trajectory. Denote by Γ and Γ' the set 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 trajectories² γ and γ' can uniquely define: (i) the spatial relation, (ii) the temporal relation, (iii) can provide a convenient residual error measure: $err(\vec{P}) = \sum_{\vec{x} \in \gamma} D(\vec{x}, \vec{x}') = \sum_{t \in [t_0, t_n]} d(\vec{p}(t), \vec{p}'(t'))$, where $[t_0, t_n]$ is the temporal support of the space-time

trajectory γ , $\vec{p}(t)$ is the spatial 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 case (2D) 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 case (3D) 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). Each step of the algorithm is then explained in more detail below:

²By 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).

- (1) Construct feature trajectories (i.e., detect and track feature points for each sequence).
- (2) For each trajectory estimate its basic properties.
- (3) Based on basic properties construct an initial correspondence table between trajectories.
- (4) Estimate candidate parameter vector $\vec{P} = (P_{spatial}, P_{temporal})$ by repeatedly choosing (at random) a pair of possibly corresponding trajectories³. At each trial compute the set of parameters \vec{P} which minimizes 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 such that their residual error is small (or the median residual error).
- (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 (the reliability of the center of mass to be used as a feature point is discussed in [10], and the reliability of extreme points on silhouettes is discussed in [18]). The KLT feature tracker [11, 17] 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) are matched against dynamic trajectories only. 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 we anticipate a significant change in appearance, shape properties of the *trajectories* could also be used (e.g., normalized length, average speed, curvature, 5-point projective invariance [12]). Although some of these are not projective invariants, they are useful for a crude initial sorting (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 $\vec{P} = (h_{11}, \dots, h_{33}, \Delta t)$, or $\vec{P} = (f_{11}, \dots, f_{33}, \Delta t)$ (where h_{11}, \dots, h_{33} or f_{11}, \dots, f_{33} are the components of a homography H , or a fundamental matrix F , respectively), we minimize the following error function⁴ (Step 4 and Step 8) :

$$\vec{P} = \underset{\vec{P}}{\operatorname{argmin}} \sum_{\gamma \in \Gamma} \sum_{t \in \operatorname{support}(\gamma)} d(\vec{p}(t), \vec{p}'(s \cdot t + \Delta t)) \quad (1)$$

where, $d()$ is either $dist_H()$ or $dist_F()$ depending whether the scene is 2D or 3D. 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., [9] 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:

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

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 fixed interval (20-25 frames in our experiments).

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

⁴In Step 4 the summation is only over the selected trajectory.

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. Beyond the increase in the data size (sequences vs. images), some degenerate cases may result in space-time ambiguities (see [3, 6, 15]). However, when the scene dynamics is rich enough, sequence matching is superior to image matching in multiple ways. Beyond providing sub-frame temporal alignment (synchronization) between video sequences⁵ (which gives rise to new video applications, see [14]) it also provides the following benefits over feature-based image matching:

(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. Insufficient common appearance information across images can also occur when the two cameras record the scene at significantly different zooms (such as in Fig. 3.a and 3.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. 4.a and 4.b), thus sensing different features in the scene. In those cases the photometric properties of the two input sequences are very different. Yet, the trajectories of moving objects over time are independent of the sensor properties, thus form a powerful cue for matching across the two sequences. In other words, the need for consistent appearance information is replaced by consistent temporal behavior, as captured by trajectories of moving objects estimated *within* each sequence separately.

If the two cameras are further known to observe similar features (e.g., when they relatively close to each other and have similar imaging properties), the regular feature tracking (e.g., KLT) may be used (for static as well as dynamic scene points). If the cameras have different sensing modalities or a wide base-line, we prefer to select “symbolic” features such as the center of gravity of the moving objects (e.g., the cars and the kite in Fig. 4) or the upper tip of moving silhouettes⁶ (e.g., the moving people or the ball in Fig. 6) as the feature points.

(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.

(iii) **Reduced Combinatorial Complexity.** When the search space for spatial correspondences is large (such as in wide base-line matching, or under large image rotations, or under large zoom differences), sequence-to-sequence matching can be used to reduce the combinatorial complexity. First, correspondence of trajectories is less ambiguous than correspondence of feature points due to the added “shape” properties of feature trajectories. This is illustrated in Fig. 2 and discussed in Section 1. Second, in these cases the number of trials required by a RANSAC-like algorithm is significantly lower

⁵The accuracy of the sub-frame temporal alignment was empirically verified, and in our experiments was found to be up to ± 0.1 frame-time.

⁶Implicitly assuming that the cameras are horizontal, and the object tip is not occluded in one camera.

in sequence-to-sequence matching. We next analyze the difference between the number of trials in both cases (image-to-image vs. sequence-to-sequence matching).

Let m be the minimal number of correspondences required for computing a spatial transformation $\vec{P}_{spatial}$. Let ϵ be the probability that a feature matching across the two images is correct (i.e., the probability that it is a mismatch or an outlier is $(1 - \epsilon)$). A RANSAC-like matching algorithm requires that at least one of the trials (i.e., one random sample of m correspondences) will not contain any mismatches (outliers). Then N – 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, is given by the following formula [13, 9]:

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

This formula emphasizes why a limited search range for candidate feature correspondences is crucial for regular image-to-image matching. A *bounded* search guarantees that ϵ is large enough (e.g., $\epsilon > 0.5$), thus limiting the number of trials N to a reasonable number.

However, when there is a large base-line between the cameras, a large scale difference, or a large image rotation, then $\epsilon \approx \frac{1}{\#features}$ (the probability to choose corresponding features at random). ϵ may be even smaller if the two sets of features from the two images are inconsistent. In such cases the contribution of sequence-to-sequence matching is most prominent. Thus, for example, when using image-to-image matching with 100 features, according to Eq. (3) the number of necessary trials for computing a homography ($m = 4, \epsilon = \frac{1}{100}, p = 99\%$) is $N > 46,000,000 = 4.6 \times 10^8$. On the other hand, when using sequence-to-sequence matching, even if we do not exploit distinguishing shape trajectory properties and still assume that $\epsilon = \frac{1}{\#trajectories}$, we get a reasonable number of trials, e.g., for $\epsilon = \frac{1}{100}$, $m = 1$, and $p = 99\%$, we get $N \geq 459$.

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).

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. Furthermore, when moving objects appear at different times in the sequence, the complexity of trajectory matching is even further reduced. However, even though there are only few *trajectory* correspondences, they still provide a high number of *point* correspondences (in our experiments trajectories typically have more than a hundred points – one per frame).

5 Applications and Results

The above mentioned benefits of sequence-to-sequence matching/alignment give rise to new video applications, that are otherwise very difficult or even impossible to obtain using existing image-to-image matching tools. Some of these are briefly mentioned below:

(1) *Recovering Large Transformations*. By large transformations we refer to cases where the search range for corresponding features is large, such as in large image rotations, or large scale (zoom) differences. An example of matching two images under large scale difference is shown in Fig. 3, where the difference in zoom between the two sequences was about 1 : 3. For more details see figure caption.

(2) *Wide baseline matching*. Figs. 5 and 6 show two examples of recovering fundamental matrices using two cameras situated on opposite sides of the scene (i.e., the cameras are facing each other). Note, that the estimated epipoles fall very close to their true image location (i.e., on the image of the other camera). For more details see figure captions.

(3) *Multi-sensor matching*. An example of multi-sensor matching is shown in Fig. 4. In this example, the cameras were located next to each other, therefore the spatial matching was modeled by a homography (see figure caption for more details). Note, that if the IR and visible light cameras were situated farther apart, then we would compute a fundamental matrix between the IR and the visible light cameras, using the same object trajectories. We are not aware of any other method that can compute fundamental matrices between cameras of different sensing modalities.

References

- [1] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV)*, pages 237–252, Santa Margarita Ligure, May 1992.
- [2] P.R. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *International Conference on Computer Vision (ICCV)*, pages 173–182, Berlin, May 1993.
- [3] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 682–689, Hilton Head Island, South Carolina, June 2000.
- [4] O. Faugeras, Q.T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.
- [5] M. A. Fischler and R.C. Bolles. Ransac random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395, 1981.
- [6] M. A. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *International Journal of Computer Vision*, 38(1):59–73, 2000.
- [7] F.R. Hampel, P.J. Rousseeuw, E. Ronchetti, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, New York, 1986.
- [8] C.G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.
- [9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, Cambridge, 2000.
- [10] L. Lee, R. Romano, and G. Stein. 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, August 2000.
- [11] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.
- [12] J.L. Mundy and A. Zisserman. Geometric invariance in computer vision. In *MIT Press*, 1992.
- [13] P. Rousseeuw. *Robust Regression and Outlier Detection*. Wiley, New York, 1987.
- [14] E. Shechtman, Y. Caspi, and M. Irani. Increasing video resolution in time and space. In *European Conference on Computer Vision (ECCV)*, Copenhagen, May 2002.
- [15] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *DARPA IU Workshop*, pages 1037–1042, Monterey CA, 1998.
- [16] R. Szeliski and H.-Y Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics Proceedings, Annual Conference Series*, pages 251–258, 8 1997.
- [17] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [18] K.-Y. K Wong and R. Cipolla. Structure and motion from silhouettes. In *International Conference on Computer Vision (ICCV)*, volume II, pages 217–222, Vancouver, Canada, July 2001.
- [19] C. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [20] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.
- [21] I. Zoghliami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 420–425, June 1997.



Figure 3: **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 were located next to each other 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 ball's center of gravity trajectory), 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.

For color sequences see attached tar file.

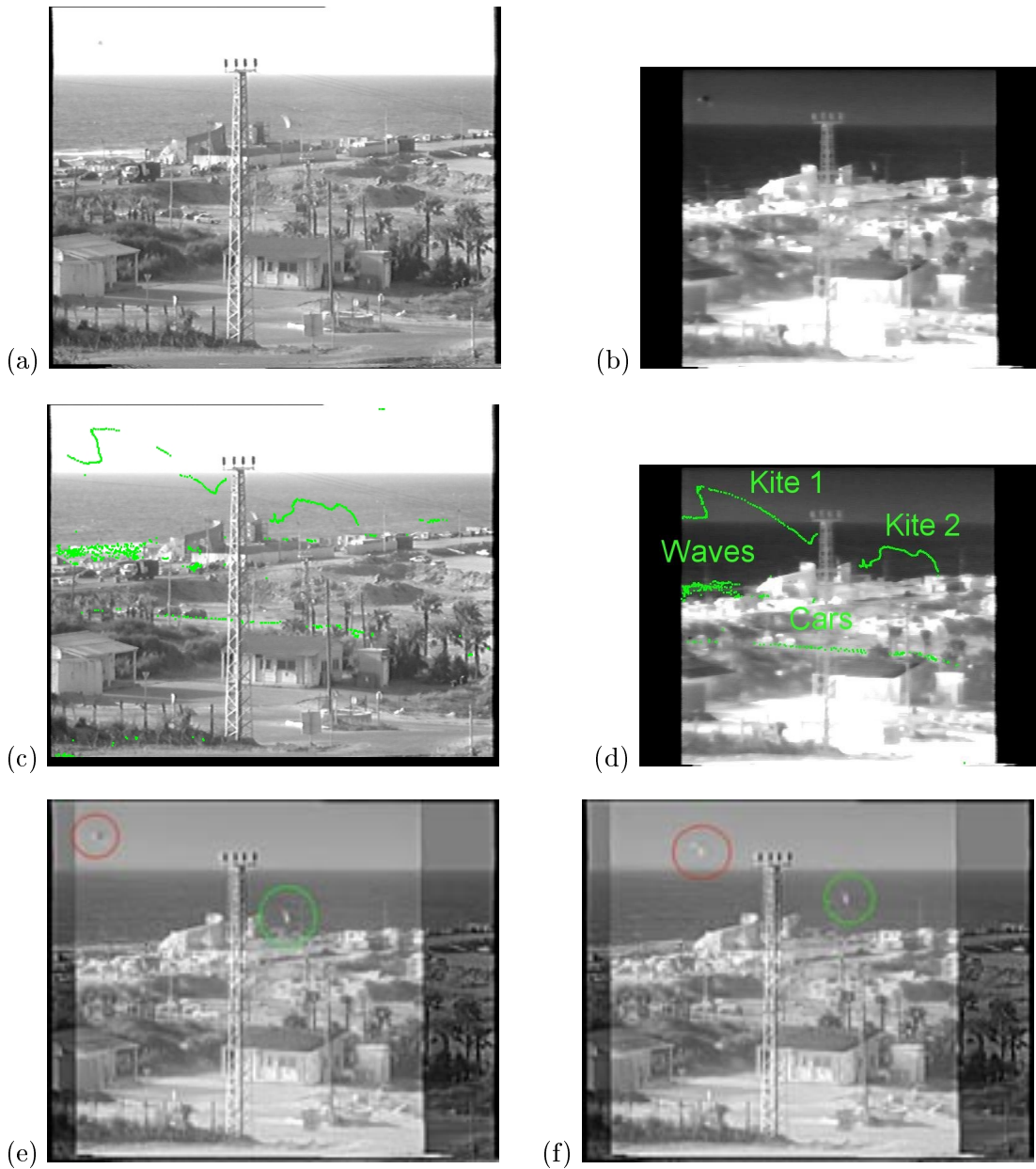


Figure 4: **Multi-Sensor Alignment.** (a) and (b) display representative frames from a PAL visible light sequence and an NTSC Infra-Red sequence, respectively. The scene contains several moving objects: 2 kites, 2 moving cars, and sea waves. The trajectories induced by tracking the moving objects are 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 [2]). 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 green), and the left kite which is more clearly visible in the IR sequence (circled in red).

(a) First camera sequence:



(b) Second camera sequence:

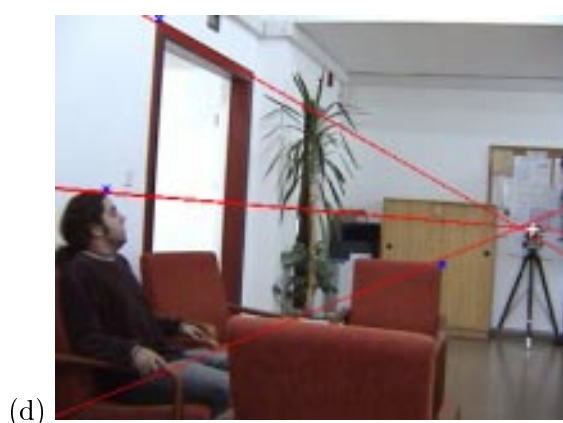
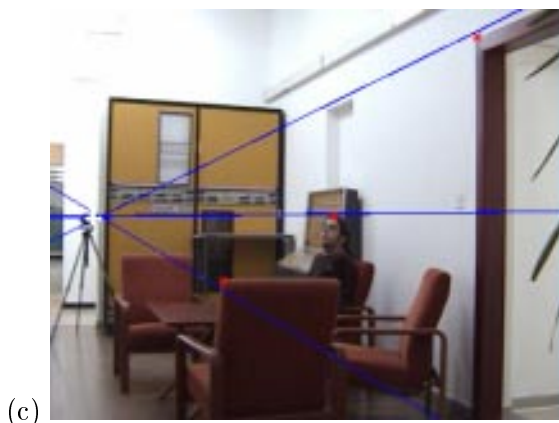


Figure 5: **Wide Base-Line Matching** Rows (a) and (b) display a few corresponding frames of one person (out of three that took part in the experiment) walking and sitting in a hall. The sequences were taken from two opposite sides of the hall. Each camera is visible by the other camera and is marked 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 -2.82 frames.



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. Each camera is visible by the other camera and is marked by a white arrow. Space-time trajectories induced by moving objects (ball and players) are displayed in (c)-(d) in different colors for the different objects. The tracked feature points that correspond to the current frame are marked in yellow. 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 distance of a 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 (the other camera).

In this example the blob size was used to provide initial correspondences (in steps 2-3 of the algorithm). Two trajectories (instead of one) were used on each RANSAC iteration (steps 4-6), 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 3.69 frames.