Spatio-Temporal Alignment of Sequences[*][†]

Yaron Caspi **Michal Irani**
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel

Email: {caspi,irani}@wisdom.weizmann.ac.il

Abstract

This paper studies the problem of sequence-to-sequence alignment, namely establishing correspondences in *time* and in *space* between two different video sequences of the same dynamic scene. The sequences are recorded by uncalibrated video cameras, which are either stationary or jointly moving, with fixed (but unknown) internal parameters and relative inter-camera external parameters. Temporal variations between image frames (such as moving objects or changes in scene illumination) are powerful cues for alignment, which cannot be exploited by standard image-to-image alignment techniques. We show that by folding spatial and temporal cues into a single alignment framework, situations which are inherently ambiguous for traditional image-to-image alignment methods, are often uniquely resolved by sequence-to-sequence alignment. Furthermore, the ability to align and integrate information across multiple video sequences both in *time* and in *space* gives rise to new video applications that are not possible when only image-to-image alignment is used.

1 Introduction

The problem of image-to-image alignment has been extensively studied in the literature ([3, 4, 19, 24, 20, 29, 33, 34] to list just a few). By *"image-to-image alignment"* we refer to the problem of estimating dense point correspondences between two or more images, i.e., for each pixel (x, y) in one image, find its corresponding pixel in the other image: $(x', y') \leftrightarrow (x + u, y + v)$, where (u, v) is the spatial displacement. This paper addresses a different problem – the problem of *"sequence-to-sequence alignment"*, which establishes correspondences both in *time* and in *space* between multiple *sequences* (as opposed to multiple images). Namely, for each pixel (x, y) at frame (time) t in one sequence, find its corresponding time t' and position (x', y') in the other sequence: $(x', y', t') = (x + u, y + v, t + w)$, where (u, v, w) is the *spatio-temporal* displacement. Note, that (u, v) (the spatial displacement) and w (the temporal displacement) are not necessarily integer values, i.e., they may be sub-pixel or sub-frame values.

There are two main motivations for using sequence-to-sequence alignment:

[*] A preliminary version of this paper appeared in CVPR' 2000 [8].

[†] This work was supported by the Moross Laboratory for Vision and Motor Control.

1. It can resolve spatial ambiguities and handle situations where image-to-image alignment fails.

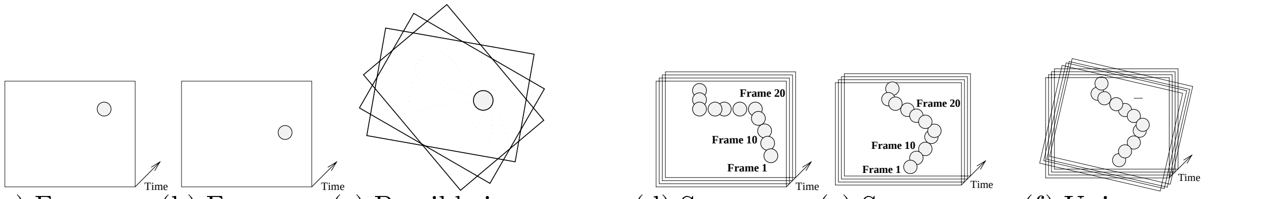
2. The ability to align and integrate information across multiple sequences both in space and in time gives rise to new video applications that are not possible when only image-to-image alignment is used.

These are briefly explained here and further elaborated in Sections 4 and 5. Image-to-image alignment methods 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 alignment. One such example is illustrated in Fig. 1. 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). Moreover, scene dynamics is not limited to moving objects. It also includes non-rigid changes in the scene (e.g., flowing water), changes in illumination, etc. All these changes are not captured by any of the individual frames, but are found *between* the frames. The scene dynamics 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 alignment across sequences.

We show in the paper (Section 4) that by folding spatial and temporal cues into a single alignment framework, situations that are inherently ambiguous for image-to-image alignment methods are often uniquely resolved by sequence-to-sequence alignment. Furthermore, in situations where there is very little common appearance (spatial) information across the two sequences, such as in alignment of sequences of different sensing modalities (e.g., Infra-Red and visible-light sensors), coherence of the scene dynamics (i.e., temporal cues) becomes the major source of information for alignment of the two sequences.

Sequence-to-sequence alignment enables integration of information across multiple video sequences. This can be used to generate new video sequences which exceed the spatial and temporal physical bounds of a single sensor. In particular, it allows to exceed the limited spatial resolution (via super-resolution, e.g., [17]), the limited depth of focus, the limited dynamic range, the limited spectral response (e.g., via fusion of multiple sensing modalities [7]), and the limited field of view. While *spatial* bounds of sensors can also be exceeded via image-to-image alignment, sequence-to-sequence alignment further allows to exceed *temporal* bounds of sensors. For example, it allows to exceed the limited temporal resolution (the limited frame rate) of recorded sequences. Temporal super-resolution allows visual observation of dynamic events that occur faster than frame-rate, and therefore cannot be seen in any of the input video sequences. Temporal super-resolution requires temporal alignment of the sequences at *sub-frame* accuracy which cannot be obtained by image-to-image alignment. This and other applications of sequence-to-sequence alignment are discussed in Section 5.

We present in the paper two possible sequence-to-sequence alignment algorithms. One is a direct gradient-based sequence-to-sequence alignment algorithm, and the other is a feature-based



(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 corresponding frames in time 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).*

sequence-to-sequence alignment algorithm. Both algorithms receive as input two video sequences and simultaneously estimate the spatial and temporal transformation between the two sequences. The current implementations assume parametric transformations in space and in time. However, the concept of sequence-to-sequence alignment is more general and is not limited to the particular algorithms or implementations described in this paper. Possible extensions of these algorithms to more complex models are also briefly sketched.

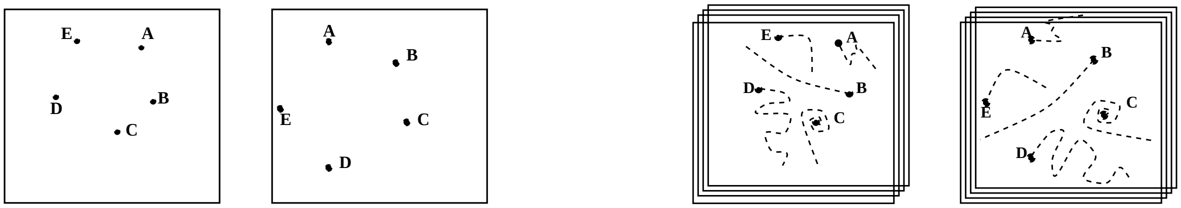
The rest of the paper is organized as follows: In Section 2 we formulate the problem of sequence-to-sequence alignment. In Section 3 we present two sequence-to-sequence alignment algorithms (the feature-based and the direct-based). Section 4 discusses the properties of sequence-to-sequence alignment, and Section 5 describes potential applications of sequence-to-sequence alignment.

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, and let $\vec{\mathbf{x}}' = (x', y', t') = (x + u, y + v, t + w)$ be the corresponding space-time point in sequence S'. The spatio-temporal displacement $\vec{\mathbf{u}} = (u, v, w)$ need not be of integer values. u,v (the *spatial* displacements) can be sub-pixel displacements, and w (the *temporal* displacement) can be a sub-frame time shift. While every space-time point $\vec{\mathbf{x}}$ has a different local spatio-temporal displacement $\vec{\mathbf{u}}$, we assume they are all *globally* constrained by a single parametric model $\vec{P} = (\vec{P}_{spatial}, \vec{P}_{temporal})$. The recorded scene can change dynamically, i.e., it can include moving objects, non-rigid deformations of the scene, changes in illumination over time, and/or other types of temporal changes. The cameras can be either stationary or jointly moving with fixed (but *unknown*) internal and relative external parameters.

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, and may be at sub-frame time units.

The *spatial misalignment* between the two sequences results from the fact that the two cameras



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

Figure 2: **Point 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).*

have different external and internal calibration parameters. In our current implementation $P_{spatial}$ was chosen to be a 2D projective transformation (homography). 2D projective transformations approximate the inter-sequence spatial transformation when the distance between the camera projection centers is negligible relative to the distance of the cameras from the scene, or if the scene is roughly planar. Note that althogh the inter-sequence transformation is a simple 2D parametric transformation, the intra-sequence changes (i.e., changes between consecutive frames) can be very complex.

Let $\vec{p} = (x, y, 1)^T$ denote the homogeneous coordinates of only the *spatial* component of a space time point $\vec{\mathbf{x}} = (x, y, t)$ in S. Let H be the 3×3 homography matrix of the *spatial* parametric transformation between the two sequences, $H = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$. Then, corresponding space-time point $\vec{\mathbf{x}}' = \vec{\mathbf{x}} + \vec{\mathbf{u}}$ can be expressed by: $x' = \frac{H_1\vec{p}}{H_3\vec{p}}$, $y' = \frac{H_2\vec{p}}{H_3\vec{p}}$, where H_i is the ith row of H, and for the temporal components by $t' = s \cdot t + \Delta t$ (1D affine transformation in time). Note that H is common to all frames because the cameras are fixed relative to each other over time (both internal parameters and inter-camera external parameters). Also, 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, the unknown parameters are: $\vec{P} = [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ h_{31}\ h_{32}\ h_{33}\ \Delta t]$, i.e., 10 unknowns with 9 d.o.f. (the homography is defined only up to scale)[1].

While in the current implementations the inter-camera spatial transformations are 2D parametric transformations, the framework presented in this paper is more general, and is not restricted to 2D transformations alone. Thus for example $\vec{P}_{spatial}$ may represent the entries of the fundamental matrix, or may be extended to other 3D models to include shape parameters, similar to the hierarchy of spatial alignment models described in [3]. $\vec{P}_{temporal}$ can also be a non-parametric transformation in time (e.g., see [11, 12]).

[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).

3 Sequence-to-Sequence Alignment Algorithms

This section proposes two possible algorithms for sequence-to-sequence alignment: A feature-based algorithm (Section 3.1), and a direct gradient-based algorithm (Section 3.2).

3.1 Feature-Based Sequence Alignment

Typical feature-based *image* alignment methods (see [31] for a review) first apply a local operator to detect interest points in a pair of images (e.g., the Harris corner detector [14]). Once interest points are extracted in the two images, robust estimation methods, such as RANSAC [10], LMS [13], etc, are used for finding corresponding points and extracting the spatial transformation between the two images. In some other cases [32] a correlation based matching is used to initialize the approximation of matching features. In general the correlation may be based on any properties of a feature point, but it is usually based on brightness values of small neighborhoods of the feature point.

Feature-based image-to-image alignment can be generalized to feature-based sequence-to-sequence alignment by extending the notion of features from *feature points* into *feature trajectories*. A feature trajectory is the trajectory of a point (static or dynamic) representing its location in each frame along the sequence. Spatio-temporal alignment between the two sequences can then be recovered by establishing correspondences between trajectories. The advantage of this approach is illustrated in Fig. 2, which 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 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 across the two sequences (i.e., which trajectory corresponds to which trajectory), as shown in Fig. 2.c and 2.d. Furthermore, a single pair of (non-trivial) corresponding trajectories (i.e., a trajectory of an object which is not moving on a straight line and covers a large enough image region) can uniquely define: (i) the spatial transformation, (ii) the temporal transformation, (iii) can provide a convenient error measure for the quality of the extracted spatio-temporal alignment.

We next outline the feature-based sequence-to-sequence alignment algorithm that we have used in our experiments (which is a RANSAC-based algorithm). 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, or other properties as explained below).

(3) Based on basic properties construct an initial correspondence table between trajectories.

(4) Estimate candidate parameter vectors $\vec{P} = (P_{spatil}, P_{temporal})$ by repeatedly choosing (at random) a pair of possibly corresponding trajectories[2]. At each trial compute the parametric spatio-temporal transformation $\vec{P}$ which best aligns the two trajectories.

(5) Assign a score for each candidate $\vec{P}$ to be the number of corresponding pairs of trajectories whose distance after alignment by $\vec{P}$ is smaller than some threshold.

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

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

(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 feature trajectories were computed either by using the KLT feature tracker [22, 30] or by tracking the center of mass of moving objects (Step 1). The trajectories were then classified as static or dynamic, to reduce the complexity of trajectory correspondences (Step 2). In the presence of many trajectories, shape properties of the trajectories may also be used (e.g., normalized length, average speed, curvature, 5-points projective invariance). Although some of these are not projective invariants, they are useful for crude initial sorting (Step 3).

Two matching trajectories across the two sequences induce multiple point correspondences across the camera views. These point correspondences are used for computing the spatial and temporal transformation between the two sequences. In our current implementation $\vec{P}_{spatial}$ is a homography. However the same framework may be used for recovering a fundamental matrix in the presence of 3D parallax (e.g., when the two video sequences are recorded from different viewpoints). A similar approach embedded in an event detection framework was taken by [28]. To evaluate a candidate transformation parameter $\vec{P} = (h_{11}, \cdots, h_{33}, \Delta t)$, where $h_{11}, \cdots, h_{33}$ are the components of a homography H, we minimize the following error function[3] (Step 4 and Step 8) :

$$\vec{P} = \underset{H, \Delta t}{\mathbf{argmin}} \sum_{Trajectories} \left(\sum_{t \in Trajectory} ||p'(s \cdot t + \Delta t) - H(p(t))||^2 \right) \tag{1}$$

where, $p(t) = [x(t), y(t), 1]^T$ is the spatial position (i.e., pixel coordinates) of a feature point along the trajectory at time t (in homogeneous coordinates), H is a homography, and $p'(s \cdot t + \Delta t)$ is the location of the corresponding feature point in the corresponding trajectory in the other sequence at time: $t' = s \cdot t + \Delta t$. Since t' is not necessarily an integer value (allowing sub-frame time shift), it is interpolated from the adjacent (integer time) point locations: $t_1 = \lfloor t' \rfloor$ and $t_2 = \lceil t' \rceil$. The minimization was performed by alternating the following two steps:
(i) Fix Δt and approximate H using standard methods (e.g., the DLT algorithm described in [15]).
(ii) Fix H and refine Δt by fitting the best linear interpolation value. In other words we search for $\alpha = t' - t_1$ such that minimizes:

$$\min_{\alpha} \sum_{t} ||(p'(t_1) \cdot (1 - \alpha) + p'(t_2) \cdot (\alpha)) - Hp(t)|| \ : \ \alpha \in [0..1]. \tag{2}$$

The iterations stop when the residual error does not change[4]. Only a few (less than 5) iterations were required in all cases. As an initial guess for the spatial transform, we used the identity homography, and performed an exhaustive search over *integer* time shifts within a given time interval.

[3] In Step 4 the summation is over only one trajectory.

[4] When the spatial model is affine (i.e., $h_{31} = h_{32} = 0$ and $h_{32} = 1$ in the homography H), it is possible to approximate the spatial and temporal parameters simultaneously (without iterations), since the spatial parameters do not multiply the unknown time parameter.

The above approach can similarly be used for estimating the fundamental matrix F between two sequences taken from separate views (i.e., in the presence of 3D parallax). Eq. (1) would then become:

$$\vec{P} = \underset{F,\Delta t}{\textbf{argmin}} \sum_{Trajectories} \Big(\sum_{t \in Trajectory} ||p'(s \cdot t + \Delta t)^T \; F \; p(t)||^2 \Big) \quad (3)$$

We currently implemented and experimented only with the homography-based version of sequence-to-sequence alignment.

Stein [26] and Lee et.al [21] 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 [26, 21] and our approach. The centroids in [26, 21] 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").

In our experiments we used two types of feature trajectories: (i) Feature points were automatically selected and tracked using the KLT package [5], and (ii) Centroids of moving objects were detected and tracked using blob tracking. In general, the suggested algorithm is not limited to a particular choice of features. The advantages of tracking centroids of moving objects are discussed in [21]. In particular they emphasize the stability and invariance of such "features" to wide base line transformations. Our experiments confirm their results. We further observed the following advantage of using trajectories of moving objects centroids over trajectories of intensity-based interest points. Multiple disparate interest points on a translating rigid object (e.g., on a large moving object) may produce similar trajectories, because they undergo the same 3D motion. This results in possible ambiguities in trajectory correspondences. Taking centroids of moving objects eliminates this problem, because each moving object is extracted as one part (and not as several). Ambiguities in trajectory matching is handled by incorporating an outlier rejection mechanism into Step 5 of the algorithm, i.e., iterative estimation of $\vec{P}$ using all trajectories supporting the current candidate, and updating the score accordingly. On the other hand, because each moving object contributes only one point per frame (the centroid), and because there may be only a small number of moving objects, the sequence length required to uniquely resolve the alignment may increase significantly (to allow coverage of a large enough image region by the moving objects). We therefore use both types of point trajectories. Robust methods other than RANSAC (see [27] for a nice review) can also be incorporated into the sequence-to-sequence alignment algorithm.

3.2 Direct-Based Sequence Alignment

The previous section focused on exploiting dynamic information that is mainly due to moving objects and requires prior detection and tracking of such objects. However, scene dynamics is not

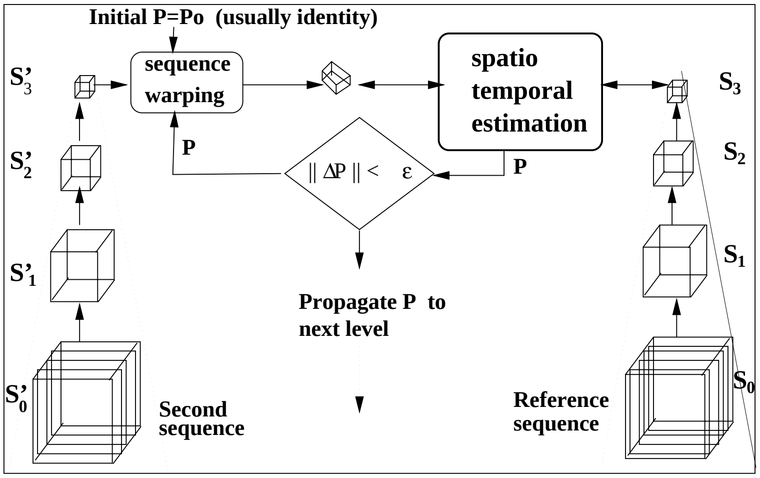


Figure 3: **Direct sequence-to-sequence alignment.** *A spatio-temporal pyramid is constructed for each input sequence: one for the reference sequence (on the right side), and one for the second sequence (on the left side). The spatio-temporal alignment estimator is applied iteratively at each level. It refines the approximation based on the residual misalignment between the reference sequence and warped version of the second sequence (warping in time and in space, marked by a skewed cube). The output of the current level is propagated to the next level to be used as an initial estimate.*

limited to moving objects. The scene may also contain more complex dynamic changes such as non rigid deformations (e.g., flowing water, flickering fire, etc.) or changes in illumination. Such changes are not conveniently modeled by feature trajectories, yet are captured by spatio-temporal brightness variations within each sequence. In this section we describe a direct intensity-based sequence-to-sequence alignment algorithm which exploits such dynamic changes.

In direct image-to-image alignment (e.g., [3, 18, 29]) the spatial alignment parameters between two *images* were recovered directly from image brightness variations. This is generalized here to recover the *spatial* and *temporal* alignment parameters between the two *sequences* directly from sequence brightness variations. The coarse-to-fine estimation framework is also generalized here to handle both time and space.

We recover the spatio-temporal displacement parameters $\vec{P}$ by minimizing the following SSD error function:

$$ERR(\vec{P}) = \sum_{\vec{\mathbf{x}}=x,y,t} (S(\vec{\mathbf{x}}) - S'(\vec{\mathbf{x}} + \vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{P})))^2. \tag{4}$$

The parameter vector $\vec{P} = \left(\vec{P}_{spatial}, \vec{P}_{temporal}\right)$ that minimizes the above error function is estimated using the Gauss-Newton minimization technique. Similar to the way it was done in [29] for image-to-image alignment, at each iteration we linearize the term in parentheses of Eq. (4) as follows (see Appendix A):

$$ERR(\vec{P}) = \sum_{\vec{x}=(x,y,t)} \left[(S(\vec{\mathbf{x}}) - S'(\vec{\mathbf{x}})) - \nabla S'^T(\vec{\mathbf{x}}) J_P \vec{P}\right]^2. \tag{5}$$

where $\nabla S'^T \ = \ [S'_x`\ S'_y\ S'_t]$ denotes the spatio-temporal derivative of the sequence S', and J_P (the Jacobian matrix) denotes the matrix of partial derivatives with respect to the unknown components of $\vec{P}$. For example, when $P_{spatial}$ is a homography, and $P_{temporal}$ is a 1D affine transformation in time, then:

$$J_P = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & x^2 & -xy & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & -xy & y^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & 1 \end{bmatrix}.$$

To recover $\vec{P}$ which minimizes Eq. (5), we differentiate $ERR(\vec{P})$ with respect to the unknown parameters of $\vec{P}$ and equate to zero. This leads to the following set of *linear equations* in $\vec{P}$, which is solved to recover $\vec{P}$:

$$\sum_{\vec{x}=x,y,t} (J_P^T \nabla S' \nabla S'^T J_P)\vec{P} = \sum_{\vec{x}=x,y,t} (S' - S) J_P^T \nabla S'. \tag{6}$$

For more details on the derivation of Eqs. (5) and (6) see Appendix A.

Because the estimation does not require detection or tracking of moving objects, nor extraction of features, it can handle very complex dynamic scenes. Note that Eq. (6) integrates all available spatio-temporal information within the sequence. Each space-time point $\vec{\mathbf{x}} = (x, y, t)$ contributes as much information as it reliably can. Any spatial or temporal variation in the scene, be it due to non-rigid motion, changes in illumination, or just a strong spatial feature in the scene, is captured by the space-time gradient $\nabla S'$, and therefore contributes to the estimation of the spatio-temporal transformation $\vec{P}$.

To allow for large spatio-temporal displacements $\vec{\mathbf{u}} = (u, v, w)$ and to speed up the convergence rate, the estimation process described above is embedded in an iterative-warp coarse-to-fine estimation framework. Fig. 3 illustrates the hierarchical spatio-temporal estimation framework. The multi-scale analysis is done simultaneously in *space* and in *time*. The Gaussian *image pyramid* [6] used in image-to-image alignment [3, 18, 29] is generalized here to a space time Gaussian *sequence pyramid*[5]. The highest resolution level in the sequence pyramid is the input sequence. Consecutive lower resolution levels are obtained by low-pass filtering the sequence at the current level both in *space* and in *time*, followed by sub-sampling by a factor of 2 in all three dimensions x, y, and t. Thus, for example, if one resolution level of the volumetric sequence pyramid contains a sequence of 64 frames of size 256 × 256 pixels, then the next resolution level contains a sequence of 32 frames of size 128 × 128, etc. In our experiments we usually employed five pyramid levels and about 5 iterations per level. The iterations were initialized by the identity transformation (i.e., no initial guess was provided).

Unlike standard 3D volumetric alignment (e.g., in medical imagery) where (x,y,z) are treated uniformly, in our case the spatial (x, y) and the temporal (t) components are of different nature. They must be treated separately, and cannot be intermixed. Furthermore, there are tradeoffs between time and space. Some of these tradeoffs are discussed in Appendix B. Although our current implementation is limited to 2D parametric spatial transformations, it can be extended to other spatial models (including 3D models), similar to the hierarchy of models described in [3] for direct image-to-image alignment.

[5] A Laplacian sequence pyramid can equally be used.

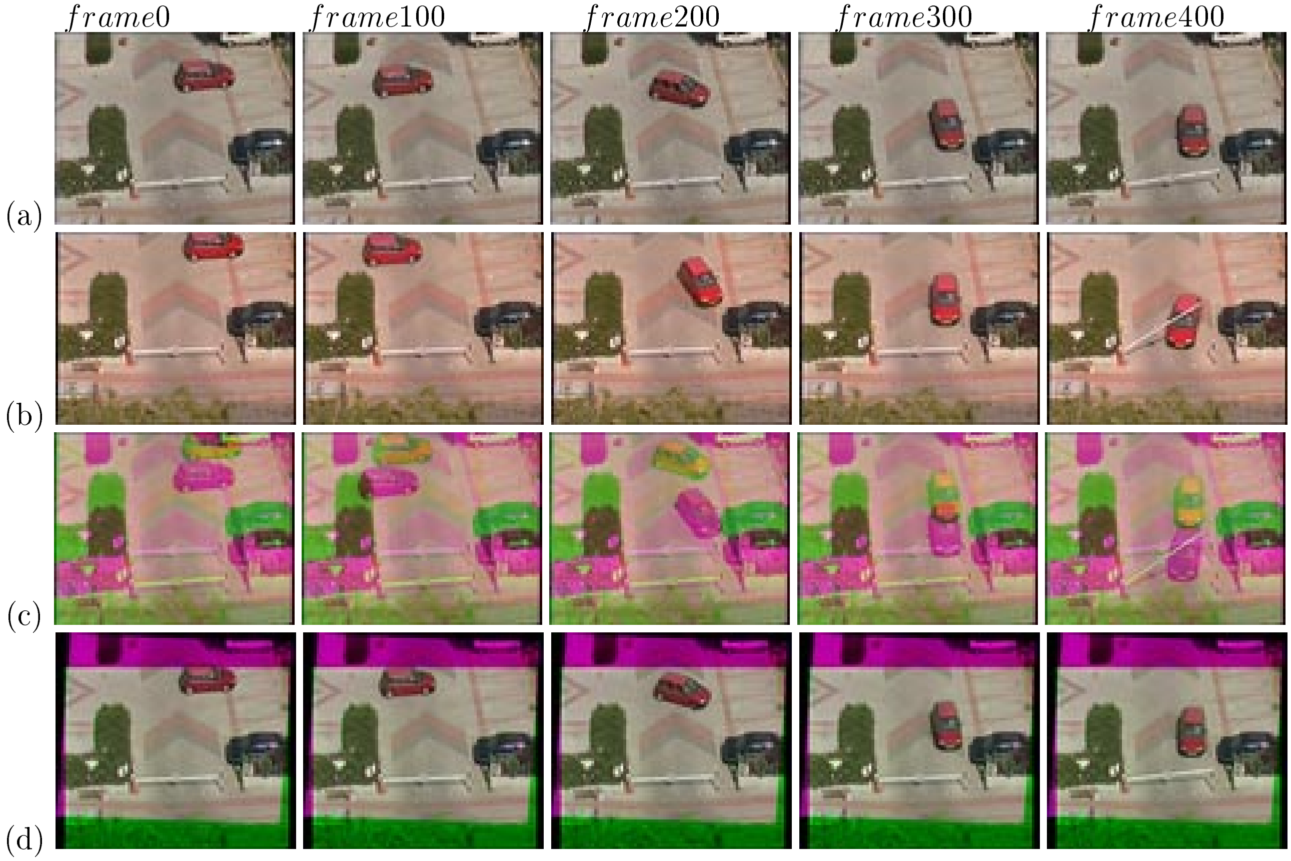


Figure 4: **Scene with moving objects.** *Rows (a) and (b) display five representative frames (0,100,200,300,400) from the reference and second sequences, respectively. The spatial misalignment is easily noticeable near image boundaries, where different static objects are visible in each sequence (e.g., the white car at the top-right portion of the frames in reference sequence (a)). The temporal misalignment is noticeable by comparing the position of the gate in frames 400: In the second sequence it is already open, while still closed in the reference sequence. Row (c) displays superposition of the representative frames* before *spatio-temporal alignment. The superposition composes the red and blue bands from reference sequence with the green band from the second sequence. Row (d) displays superposition of corresponding frames* after *spatio-temporal alignment. The dark pink boundaries in (d) correspond to scene regions observed only by the reference camera. The dark green boundaries in (d) correspond to scene regions observed only by the second camera.* **For full color sequences see www.wisdom.weizmann.ac.il/Seq2Seq.**

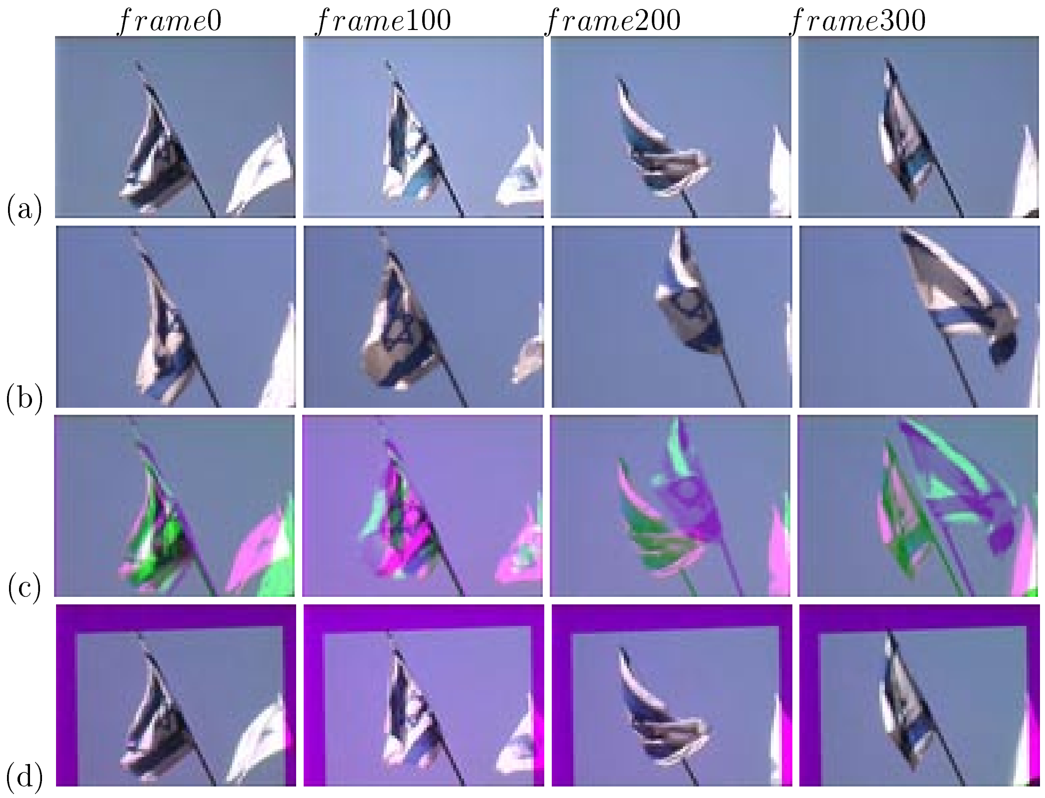


Figure 5: **Scene with non rigid motion.** *Rows (a) and (b) display four representative frames (0,100,200,300) from the reference and second sequences, respectively. Row (c) displays superposition of the representative frames* before *spatio-temporal alignment. The spatial misalignment between the sequences is primarily due to differences in cameras focal lengths (i.e., differences in scale). The temporal misalignment is most evident in frames 300.a vs. 300.b, where the wind blows the flag in opposite directions. Row (d) displays superposition of corresponding frames* after *spatio-temporal alignment, using the direct-based algorithm of Section 3.2.* **For full color sequences see www.wisdom.weizmann.ac.il/Seq2Seq.**

3.3 Examples

Before proceeding to studying properties, benefits and applications of sequence-to-sequence alignment, we show some results of applying the two proposed algorithms on real world sequences. Fig. 4 shows a scene with a car driving in a parking lot. The two input sequences Fig. 4.(a) and Fig. 4.(b) were taken from two different windows of a tall building. No synchronization between the two sequences was used. Typical sequence length is several hundreds of frames. Fig. 4.(c) displays superposition of representative frames, generated by mixing the red and blue bands from the reference sequence with the green band from the second sequence. This demonstrates the initial misalignment between the two sequences, both in time and in space. Note the temporal misalignment of dynamic objects (e.g., different timing of the gate being lifted), and spatial misalignment of static scene parts (such as the parked car or the bushes). Fig. 4.(d) shows the superposition *after* applying spatio-temporal sequence alignment. The second sequence was spatio-temporally warped towards the reference sequence according to the computed parameters.

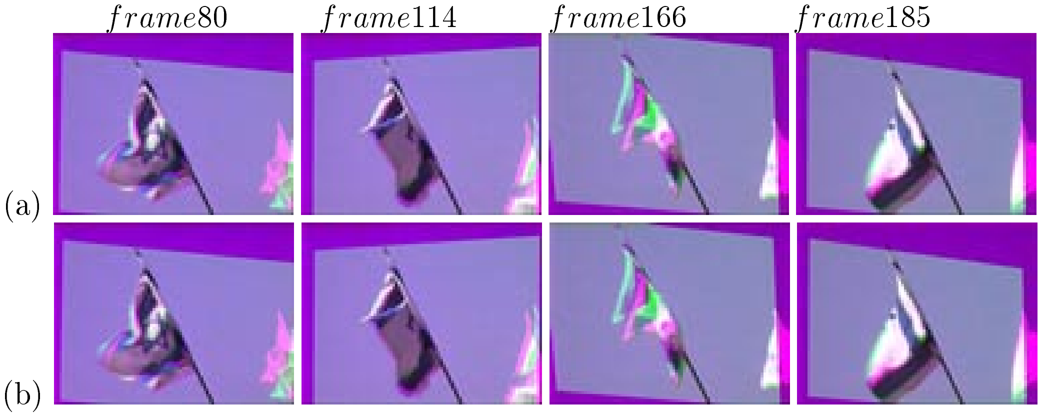


Figure 6: **Image-to-Image alignment vs. Sequence-to-Sequence alignment** *(a) Results of applying image-to-image alignment to temporally corresponding frames. Spatial alignment is inaccurate due to insufficient spatial information in any of these individual frames. (b) Accurate alignment of the same frames obtained by sequence-to-sequence alignment. The input sequences are displayed in Fig 5.*

The recovered spatial transformation indicated that the initial spatial misalignment between the two input sequences was on the order of a 1/5 of the image size, including a small rotation, a small scaling, and a small skew (due to different aspect ratios of the two cameras). The recovered temporal shift between the two sequences was 46.63 frames. Comparable results were obtained for this sequence when using both the direct sequence-to-sequence alignment (Section 3.2) and the feature-based sequence-to-sequence alignment (Section 3.1).

The example in Fig. 4 is rich in spatial texture. Image-to-image alignment therefore also provides high quality *spatial* alignment in this case (when applied to corresponding frames in time across the two sequences). However, this is not the case for the next example. Fig. 5 shows two sequences (5.a and 5.b) of a flag blowing in the wind (non-rigid motion). The spatial texture in each frame is concentrated in a small image region. Fig. 5.c shows a superposition of representative frames from both sequences *before* spatio-temporal alignment, displaying initial misalignment in time and space. Fig. 5.d shows superposition of corresponding frames *after* spatio-temporal sequence alignment (using the direct algorithm of Section 3.2). The recovered temporal shift was 31.43 frames. Empirical evaluation of the accuracy of our direct sequence-to-sequence algorithm (which was found in our experiments to be up to 0.1 sub-pixel accuracy and 0.1 sub-frame accuracy) can be found in Appendix C. More results of sequence-to-sequence alignment will be shown in Sections 4 and 5 in the context of properties, benefits and applications of sequence-to-sequence alignment.

4 Properties of Sequence-to-Sequence Alignment

4.1 Benefits of Sequence Alignment over Image Alignment

When there are no dynamic changes in the scene, then sequence-to-sequence alignment reduces to image-to-image alignment (with improved signal-to-noise ratio; see Appendix D). However, when the scene is dynamic, sequence alignment is superior to image alignment in multiple ways. Beyond providing temporal alignment, it also provides the following benefits to spatial alignment:

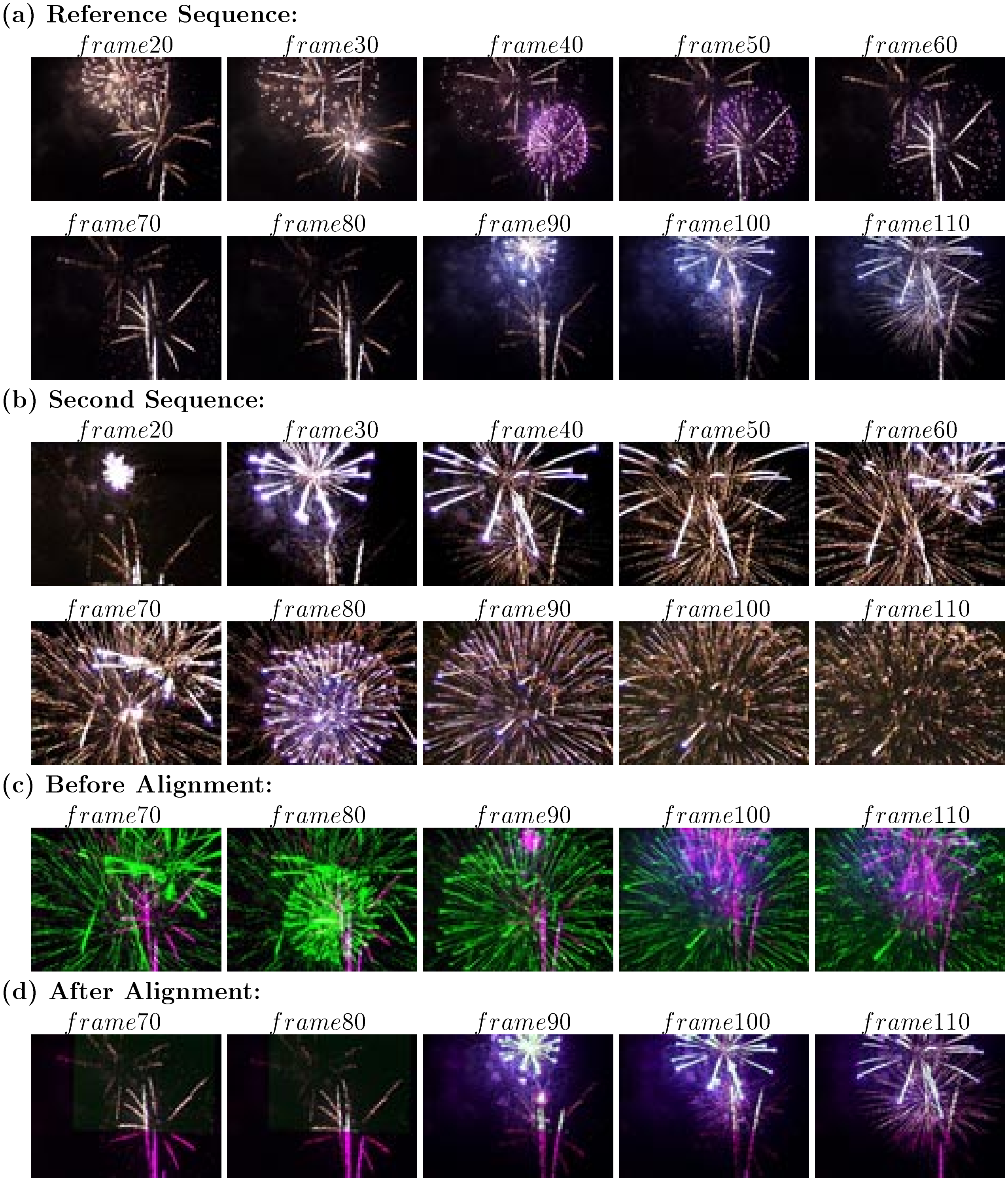


Figure 7: **A scene which constantly changes its appearance.** *Rows (a) and (b) display 10 frames (20,...,110) from the reference and second sequences of fireworks, respectively. It is difficult to visually establish the connection between the two sequences. The event in frames 90-110 in the reference sequence (7.a), is the same as the event in (approximately) frames 20-40 in the second sequence (7.b). Row (c) displays superposition of the representative frames* before *spatio-temporal alignment. The fireworks apper green and pink due to the spatio-temporal misalignment between the sequences. The spatial misalignment is mainly due to scale differences. Row (d) displays superposition of corresponding frames* after *spatio-temporal alignment, using the direct-based algorithm of Section 3.2. Due to the scale difference (approximately* 1 : 2*) there is an overlap between the two sequences only in the upper right region of every frame. Fireworks in the overlapping regions appear white, as they should. Fireworks in the non-overlapping regions appear dark pink, as they were observed by only one camera. The recovered temporal misalignment was 66.40 frames.* **For full color sequences see www.wisdom.weizmann.ac.il/Seq2Seq.**

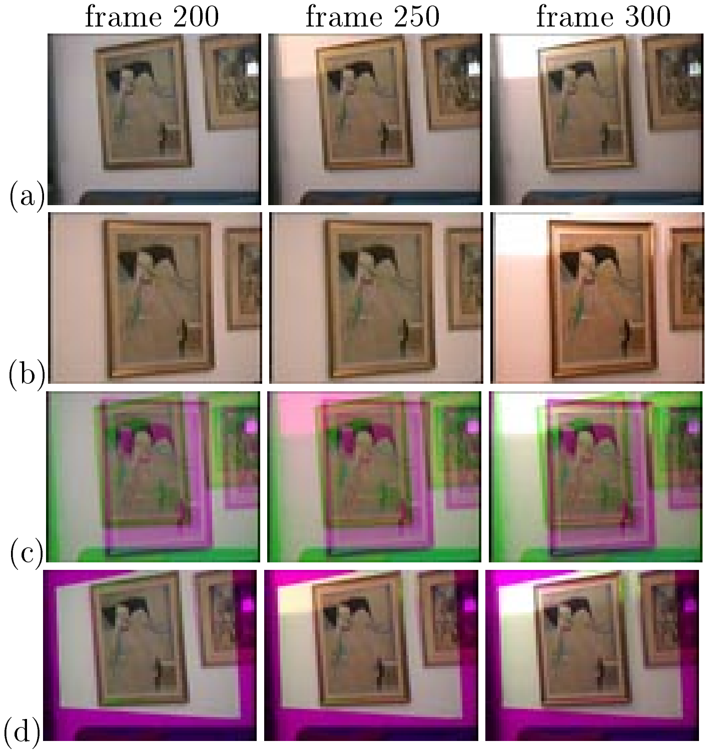
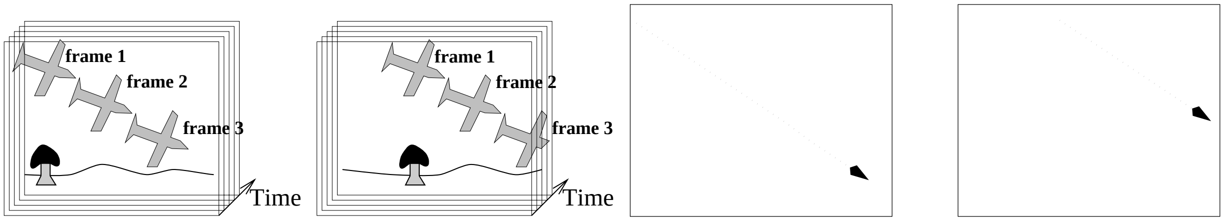


Figure 8: **Scene with varying illumination.** *Rows (a) and (b) display three representative frames (200,250,300) from the reference and second sequences, respectively. The temporal misalignment can be observed at frame 250, by small differences in illumination. (c) displays superposition of the representative frames* before *alignment (red and blue bands from reference sequence and green band from the second sequence). (d) displays superposition of corresponding frames* after *spatio-temporal alignment, using the direct-based algorithm of Section 3.2. The accuracy of the temporal alignment is evident from the hue in the upper left corner of frame 250, which is pink before alignment (frame 250.c) and white after spatio-temporal alignment (frame 250.d). The dark pink boundaries in (d) correspond to scene regions observed only by the reference camera.* **For full color sequences see www.wisdom.weizmann.ac.il/Seq2Seq.**

(i) **Resolving Spatial Ambiguities.** Inherent ambiguities in image-to-image alignment 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, or in the example shown in Fig. 6. Fig. 6 shows a comparison of image-to-image and sequence-to-sequence alignment for the input sequences of Fig. 5 (the flag blowing in the wind sequences). Image-to-image alignment performs poorly in this case, even when applied to *temporally corresponding frames*, as there is not enough spatial information in many of the individual frames. Since in this example the detected temporal misalignment (using sequence-to-sequence alignment) was $31.43 \approx 31.5$, we matched odd fields from one camera with even fields from the second camera to provide the best possible temporal correspondence for image-to-image alignment. Only 55% the of corresponding frames converged to accurate spatial alignment. The other 45% suffered from noticeable spatial misalignment. A



(a) Sequence S_1 (b) Sequence S_2 (c) Trajectory 1 (d) Trajectory 2

Figure 9: **Spatio-temporal ambiguities** *This figure shows a small airplane crossing a scene viewed by two cameras. The airplane trajectory does not suffice to uniquely determine the alignment parameters. Arbitrary time shifts can be compensated by appropriate spatial translation along the airplane motion direction. Sequence-to-sequence alignment, on the other hand, can uniquely resolve this ambiguity, as it uses both the scene dynamics (the plane at different locations) and the scene appearance (the static ground). Note that spatial information alone does not suffice either in this case.*

few representative frames (out of the 45% misaligned pairs) are shown in Fig. 6.a. These pairs of frames (as well as all the other pairs) were well aligned by sequence-to-sequence alignment (Fig. 6.b).

Insufficient common appearance information across images can also occur when the two cameras are at significantly different zooms (such as in Fig. 12) 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 10), thus sensing different features in the scene. In all these cases, the lack of common appearance information makes the problem of image-to-image alignment very difficult. However, in sequence-to-sequence alignment the need for coherent appearance information can be replaced by coherent temporal behavior, e.g., as captured by trajectories of moving objects estimated *within* each sequence separately. An example of successfully applying sequence-to-sequence alignment to such cases where image-to-image alignment is extremely difficult are shown in Figs. 12 and 10 (using the feature-based sequence-to-sequence alignment algorithm of Section 3.1). These are discussed in more detail in the “Applications” section (Sections 5.2 and 5.3).

(ii) **Improved Accuracy of Alignment.** Even when there is sufficient spatial information within the images and accurate temporal synchronization is known between the two sequences, direct sequence-to-sequence alignment may still provide higher accuracy in the estimation of the spatial transformation than image-to-image alignment. 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 alignment is restricted to alignment of existing physical frames, whereas these may not have been recorded at exactly the same time due to (possibly known) *sub-frame* temporal misalignment between the two sequences. Sequence-to-sequence alignment, on the other hand is not restricted to physical (“integer”) image frames. Because sequence warping here is done not only in space but also in time (see Fig. 3), 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 alignment. This is best illustrated by Fig. 7. The sequences show explosions of fireworks. The fireworks change their appearance (size, shape, color and brightness) drastically throughout the sequence. These rapid changes cause significant differences between "corresponding" frames in time across the two sequences, due to the residual sub-frame temporal misalignment (in this case the extracted time shift was 66.40 frames). Thus, many of these small bright dots cannot be accurately matched across physical image frames. Direct sequence-to-sequence alignment (Section 3.2), on the other hand matches elongated space-time traces of lights and not isolated spatial points of lights. The sub-frame temporal accuracy provided be sequence-to-sequence alignment is thus essential for recovering accurate sub-pixel spatial alignment.

(iii) **Reduced Combinatorial Complexity.** Another benefit of feature-based sequence-to-sequence alignment is that it significantly reduces the combinatorial complexity of feature matching, thus simplifying the correspondence problem for feature-based image alignment. There are two reasons for this: (a) Correspondence of feature 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 3.1. (b) The number of trials required by a RANSAC-like algorithm is significantly lower in sequence-to-sequence alignment. This is because the number of trials grows exponentially with the number of features to be matched. The number of feature correspondences required to compute a candidate parameter vector (e.g., a homography) in image-to-image alignment is four (4 feature points), while the number of required feature correspondences in sequence-to-sequence alignment is one (1 feature trajectory). A trajectory contains many feature points which are sorted in time. Thus, matching one point in one trajectory to another point in another trajectory automatically determines all other point correspondences across the two trajectories. One might claim that generating the trajectories involves additional computations. However, tracking is considered a much simpler problem than establishing correspondences across separate views because of its very limited search range. These additional computations are thus negligible. Note that when all feature points along a trajectory are treated as an unordered cloud of points (as in [26, 21]), there is no reduction in the complexity.

4.2 Space-Time Ambiguities

We showed how *spatial ambiguities* can often be uniquely resolved by sequence-to-sequence alignment. However, adding the temporal dimension may sometimes introduce spatio-temporal ambiguities. This occurs when different temporal alignment can compensate for different spatial alignment, and is illustrated in Fig. 9. When only the trajectory of the moving object is considered (i.e., the trajectory of the airplane), then for any temporal shift there exists a different consistent spatial transformation between the two sequences which will bring the two trajectories in Figs. 9.c and 9.d into alignment. Namely, in this scenario, using temporal changes alone provides infinitely many valid spatio-temporal transformations. Stein [26] noted this spatio-temporal ambiguity and reported its occurrence in car-traffic scenes where all the cars move in the same direction with similar velocities. Giese and Poggio [11, 12] (who modeled biological motion patterns using linear combinations of prototypical sequences) also reported a similar problem. Such ambiguities are resolved when there exists another object moving in a different direction, at a different speed, or by combining also static information (i.e., "moving objects" with zero speed).

While using information from the trajectory of the moving object alone provides infinitely many valid spatio-temporal transformations for the scenario in Fig. 9, only one of those spatio-temporal transformations is consistent with the *static background* (i.e., the tree, the horizon) or any other independent motion.

4.3 Feature-Based vs. Direct-Based Sequence Alignment

All the pros and cons of feature-based versus direct-based methods for image alignment (see [31, 16] and debate) apply here as well. However, there are additional differences between these two classes of methods that are unique to sequence alignment, because of the added temporal dimension. These are briefly discussed next.

The suggested approach to feature-based sequence alignment (Section 3.1) focuses on exploiting dynamic changes which are due to moving objects or moving points. It further requires detection and tracking of such objects. The direct approach to sequence alignment (Section 3.2), on the other hand, requires no detection or tracking of moving objects. It captures dynamic changes via the temporal derivatives without needing to explicitly model these changes by features. It can therefore handle much more complex scene dynamics, such as varying illumination (Fig. 8), non-rigid motions (Figs. 5 and 7). Moreover, a dimming or a brightening of a light source can provide sufficient information to determine the temporal alignment between the two sequences. Since global changes in illumination produce prominent temporal derivatives, even homogeneous image regions contribute temporal constraints to the direct sequence-to-sequence alignment. This is illustrated in Fig. 8. A light source was brightened and then dimmed, resulting in observable illumination variations in the scene. The effects of illumination are particularly evident in the upper left corner of the image. (Note the difference in illumination in frame 250 of the two sequences: frame 250.a and frame 250.b). The recovered temporal offset in this case was 21.32 frames. The correctness of the temporal alignment is evident from the hue in the upper left corner of frame 250, which is pink before alignment (frame 250.c) and white after temporal alignment (frame 250.d).

The limitation of the feature-based sequence alignment method in processing complex temporal changes is a result of the way the features are currently selected and tracked in the algorithm of Section 3.1. Although trajectories of features capture dynamic information, the features themselves are still 2D features within images. However, the notion of "features" can be extended from 2D features within images, to 3D space-time features within the space-time sequence volume. This will allow to capture more complex dynamic changes other than moving objects. However, appropriate volumetric spatio-temporal feature detectors must first be designed in order to obtain such a goal. Such a task is beyond the scope of this paper.

While our feature-based approach to sequence-to-sequence alignment cannot handle complex dynamic changes *within* the sequence, it can handle complex appearance changes *across* sequences, such as in sequences obtained by cameras of different sensing modalities (see Fig. 10), or cameras at significantly different zooms (e.g., 1 : 3 as in Fig. 12). In those cases the photometric properties of the two input sequences are very different. Yet, the trajectories of moving objects over time are very similar, thus forming a powerful cue for alignment across the two sequences in the feature-based alignment method. This is not the case for the direct-based alignment algorithm, which minimizes the SSD (Sum of Square Differences) between the two sequences, thus implicitly assuming similar photometric properties.

5 New and Emerging Application

Sequence-to-sequence alignment gives rise to new video applications, that are otherwise very difficult or else impossible to obtain using existing image-to-image alignment tools. These are discussed next.

5.1 Super-Resolution in Time and Space

In image-based (i.e., spatial) super-resolution [17], multiple low-resolution images (imaged at sub-pixel shifts) are combined to obtain a single high-resolution image which contains spatial features not visible in any of the input sequences. Such applications are naturally also supported by sequence-to-sequence alignment. However, beyond that, sequence-to-sequence alignment also provides *temporal* alignment at high *sub-frame accuracy.* This gives rise to totally new video applications, such as *super-resolution in time.* By super-resolution in time we mean integrating information from multiple video sequences (recorded at sub-frame time shift) into a single new video sequence of higher frame-rate (i.e., higher temporal resolution). Such a sequence can display dynamic events that occur faster than regular video frame-rate, and are therefore not visible (or else observed incorrectly) in all the input video sequences. For example, when a wheel is turning fast, beyond a certain speed it will appear to be rotating in the wrong direction in all the input video sequences (the "wagon wheel effect"). This visual effect is due to temporal aliasing. Playing the recorded video in "slow motion" will not make this effect go away. However, the reconstructed high-resolution sequence will display the correct motion of the wheel. It is interesting to note that temporal super-resolution cannot be obtained when the video cameras are synchronized using dedicated hardware (e.g., genlock). In this case all the synchronized cameras will capture the same time instance. Sequence-to-sequence alignment can therefore provide the basis for exceeding the temporal and spatial resolution of existing video cameras. For more details see [25].

5.2 Multi-Sensor Alignment

Images obtained by sensors of different modalities, e.g., IR (Infra-Red) and visible light, can vary significantly in their appearance. Features visible in one image may barely be visible in the other image, and vice versa. This poses a problem for image alignment methods. However, when trajectories of moving objects are used as the features to match across the two sequences (see Section 3.1), then the similar image appearance across the two sensors is no longer necessary. The need for coherent appearance information is replaced with coherent dynamic behavior of feature trajectories. Fig. 10 illustrates alignment of a PAL visible light sequence with an NTSC Infra-Red sequence using the feature-based algorithm of Section 3.1 with trajectories of centroids of moving objects (the two kites, waves, and several cars shown in Fig. 10.c). The differences in appearance of the objects across the two sequences will not affect the processing, which is not the case in feature-based image-to-image alignment. The results after spatio-temporal alignment are displayed after fusing the two sequences (using Burt's fusion algorithm [7]). The fused sequence clearly displays features from both sequences (representative frames shown in Fig. 10.d and 10.e).

5.3 Recovering Large Transformations and Wide Baseline Matching

Alignment of images taken at significantly different internal or external camera parameters (e.g., a wide baseline between the cameras, significant scale differences, large image rotations, etc.) is difficult. This is best understood by analyzing the number of trials that are required in a RANSAC-like algorithm to ensure accurate alignment.

Let m be the minimal number of correspondences required for computing a spatial transformation $P_{spatial}$. For example, for homography (which has 8 d.o.f) the number of required point correspondences for image-to-image alignment is $m = 4$. Let e 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 alignment algorithm requires that at least in 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 of mismatches, is given by the following formula [23, 15]:

$$N \geq \frac{log(1-p)}{log(1-e^m)}. \tag{7}$$

In regular feature-based image alignment, an initial *bounded* search for corresponding feature points is performed, to guarantee that e is large enough (e.g., $e > 0.5$), thus limiting the number of trials N to a reasonable number. However, when there is a large baseline between the cameras, a large scale difference, or a large image rotation, then $e \approx \frac{1}{\#features}$ (the probability to choose corresponding features at random). e may even be smaller if the two sets of features from the two images are inconsistent. Thus for example, if there are 100 features in the image (all appearing in both images), then according to Eq. (7) the number of necessary trials for computing a homography $(m = 4, e = \frac{1}{100}, p = 99\%)$ is $N > 46,000,000 = 4.6 \times 10^8$.

On the other hand, when using feature-based sequence-to-sequence alignment (Section 3.1), a single feature trajectory (e.g., a trajectory generated by a moving object which covers a large enough image region) suffices for computing $P_{spatial}$. This is because all point correspondences can be extracted from a single trajectory matching across the two sequences. The RANSAC-like feature-based sequence-to-sequence alignment algorithm therefore requires that at each trial only *one* feature trajectory will be matched correctly (i.e., $m = 1$). Even if we ignore the shape properties of feature trajectories and assume that all trajectories are equally likely (i.e., $e = \frac{1}{\#trajectories}$), we still get reasonable number of trials even for large transformations and baselines. For example, using Eq. (7) with $e = \frac{1}{100}$, $m = 1$, and $p = 99\%$, we get that the number of required trials is $N \geq 459$. In practice, the actual needed number of trails N is lower, because the nature of the trajectories can still be used for reliable initial matching (i.e., their shape properties or the fact that they result from static or dynamic points), thus increasing the value of e.

An example of alignment of sequences obtained at significantly different zooms (1 : 3) using the feature-based algorithm of Section 3.1 is shown in Fig. 12.

6 Conclusion and Future Work

In this paper we studied the problem of aligning two video sequences in time and in space by utilizing spatio-temporal information contained in the space-time volumes. We showed that there are several benefits to using sequence-to-sequence alignment. Since (i) it resolves many of the inherent difficulties associated with image-to-image alignment, and (ii) it gives rise to new video applications. We showed that in particular sequence-to-sequence alignment facilitates super-resolution in time, multi-sensor alignment and wide-baseline matching. We presented two specific algorithms: a direct-based sequence-to-sequence alignment algorithm, and a feature-

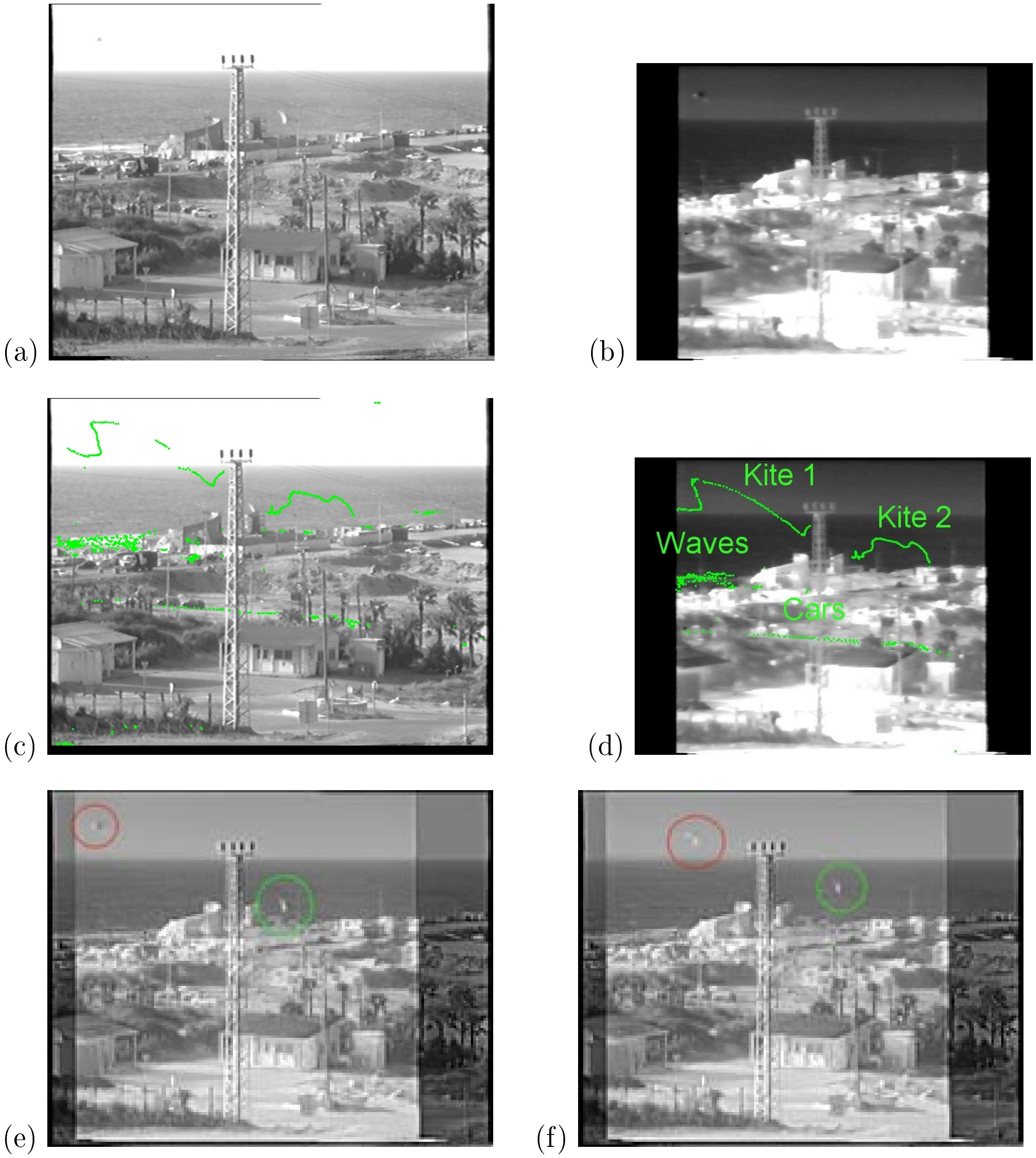


Figure 10: **Multi-Sensor Alignment.** *(a) and (b) display representative frames from a PAL visible light sequence and an NTSC Infra-Red sequence, respectively. The scene contains several moving objects: 2 kites, 2 moving cars, and sea waves. The trajectories induced by tracking the moving objects are 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 trajectory matching (Section 3.1) 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 [7]). 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).*

based sequence-to-sequence alignment algorithm. However, the notion of sequence-to-sequence alignment goes beyond the proposed algorithms in Section 3, and extends to more complex transformations in time and in space. Furthermore, sequence-to-sequence alignment can exploit not only common dynamic behavior in the scene, but also common dynamic behavior of the cameras. This gives rise to alignment of *non-overlapping* sequences [9].

References

[1] S. Baker, F. Dellaert, and I. Matthews. Aligning images incrementally backwards. Technical Report CMU-RI-TR-01-03, CMU, 2001.

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, Hawaii, December 2001.

[3] 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.

[4] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. A three frame algorithm for estimating two-component image motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 14:886–896, September 1992.

[5] S. Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. *http://vision.stanford.edu/ birch/klt*, 1996.

[6] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, 31:532–540, 1983.

[7] 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.

[8] 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.

[9] Y. Caspi and M. Irani. Alignment of non-overlaping sequences. In *International Conference on Computer Vision (ICCV)*, volume II, pages 76–83, Vancouver, Canada, 2001.

[10] M. A. Fischler and R.C. Bolles. Ransac random sample concensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395, 1981.

[11] M. A. Giese and T. Poggio. Recognition and synthesis of biological motion patterns by linear combination of prototypical motion patterns. In N. Elsner and U. Eysel, editors, *Goettingen Neurobiology Report*. Thieme Verlag, Stuttgart, 1999.

[12] 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.

[13] 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.

[14] C.G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[16] M. Irani and P. Anandan. About direct methods. In *Vision Algorithms Workshop*, pages 267–277, Corfu, 1999.

[17] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53:231–239, May 1991.

[18] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, Santa Margarita Ligure, May 1992.

[19] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16, February 1994.

[20] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(3):268–272, March 1997.

[21] 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.

[22] 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.

[23] P. Rousseeuw. *Robust Regression and Outlier Detection.* Wiley, New York, 1987.

[24] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 450–456, 1997.

[25] E. Shechtman, Y. Caspi, and M. Irani. Increasing video resolution in time and space. In *ECCV*, Copenhagen, 2002.

[26] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *DARPA IU Workshop*, pages 1037–1042, Monterey CA, 1998.

[27] C. Stewart. Robust parameter estimation in computer vision. *SIAM-Review*, 41(3):513–537, 1999.

[28] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi. Recognition action events from multiple viewpoints. In *Proc. IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[29] R. Szeliski and H.-Y Shum. Creating full view panoramic image mosaics and environments maps. In *Computer Graphics Proceedings, Annual Conference Series*, pages 251–258, 8 1997.

[30] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[31] P.H.S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *Vision Algorithms Workshop*, pages 279–29, Corfu, 1999.

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

[33] 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.

[34] I. Zoghlami, 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.

Appendix A: Derivation of the Direct Method Equations

We follow the formulation proposed in [29] for image alignment and derive the normal equations from our error function of Eq.(4):

$$ERR(\vec{P}) = \sum_{\vec{\mathbf{x}}=(x,y,t)} (S(\vec{\mathbf{x}}) - S'(\vec{\mathbf{x}} + \vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{P})))^2.$$

We linearize $S'(\vec{\mathbf{x}} + \vec{\mathbf{u}})$ using a first order Taylor approximation of S' around P_0 – the parameter vector corresponding to the identity transformation (i.e., no displacement in time or in space):

$$S'(\vec{\mathbf{x}} + \vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{P})) = S'(\vec{\mathbf{x}} + \vec{\mathbf{u}}(\vec{\mathbf{x}}; P_0)) + \nabla S'^T(\vec{\mathbf{x}'}) J_P(\vec{P} - \vec{P}_0) + \epsilon \tag{8}$$

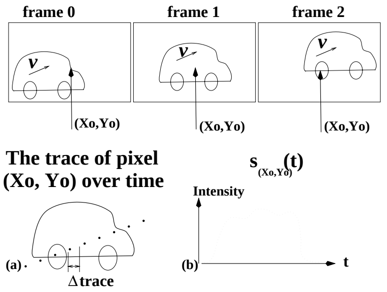


Figure 11: **Induced temporal frequencies.** *Three frames 0,1,2 of a car moving up right with velocity v are presented above. A fixed pixel* (x_0, y_0) *is marked on each frame. (a) displays the trace of the pixel. (b) displays the gray level values along this trace.*

where $\nabla S'^T = [S'_{x'} S'_{y'} S'_{t'}]$ denotes the spatio-temporal derivative of the sequence S' at $\vec{\mathbf{x}}' = \vec{\mathbf{x}} + \vec{\mathbf{u}}(\hat{\mathbf{P}})$, and $\hat{P}$ is the estimate of $\vec{P}$ from the previous iteration. J_P - the Jacobian matrix - denotes the matrix of partial derivatives of the displacement vector $\vec{\mathbf{u}} = (u, v, w)$ with respect to the components of $\vec{P}$. (Alternatively, we can linearize the term in Eq. (4) with respect to $\vec{\mathbf{x}}$, instead of with respect to the parameters $\vec{P}$, and then express the spatio-temporal displacement $\vec{\mathbf{u}}$ in terms of the parameters $\vec{P}$, similar to the way it was done for image-to-image alignment in [18] (for this formulation and its derivations see [8]).

Using the fact that $\vec{\mathbf{u}}()$ is zero at the identity transformation P_0 we obtain:

$$ERR(\vec{P}) = \sum_{\vec{\mathbf{x}}=(x,y,t)} \left[(S(\vec{\mathbf{x}}) - S'(\vec{\mathbf{x}})) - \nabla S'^T(\vec{\mathbf{x}}) J_P \vec{P} \right]^2 . \tag{9}$$

Solving the above least squares problem leads to the following set of linear equations in the unknown $\vec{P}$:

$$\sum_{\vec{\mathbf{x}}} (J_P^T \nabla S' \nabla S'^T J_P) \vec{P} = \sum_{\vec{\mathbf{x}}} (S' - S) J_P^T \nabla S' . \tag{10}$$

For computing the Jacobian matrix for the case when $P_{spatial}$ is a homography and $P_{temporal}$ is a 1D affine transformation, at each iteration we used the instantaneous approximation of a homography [3] and get:

$$J_P = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & x^2 & -xy & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & -xy & y^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & 1 \end{bmatrix} .$$

Using the formulation derived in Eq. (10), the derivatives of $\nabla S'$ must be recomputed at every iteration as S' is warped. To speed the estimation process, we can replace $\nabla S'$ by ∇S with some small modifications (which introduce an additional approximation). The same trick was proposed for image-to-image alignment in [3], and is described in further detail in [1, 2].

Appendix B: Spatio-Temporal Aliasing

This appendix discusses the tradeoff between temporal aliasing and spatial resolution. The intensity values at a given pixel (x_0, y_0) along time induce a 1-D temporal signal: $s_{(x_0,y_0)}(t) =$

$S(x_0, y_0, t)$. Due to the object motion, a fixed pixel samples a moving object at different locations, denoted by the "trace of pixel (x_0, y_0)". Thus temporal variations at pixel (x_0, y_0) are equal to the gray level variations along the trace (See Fig. 11). Denote by $\Delta trace$ the spatial step size along the trace. For an object moving at velocity v: $\Delta trace = v\delta t$, where δt is the time difference between two successive frames ($\delta t = \frac{1}{frame\ rate}$). To avoid temporal aliasing, $\Delta trace$ must satisfy the Shannon-Whittaker sampling theorem: $\Delta trace <= \frac{1}{2\omega}$, where ω is the upper bound on the spatial frequencies. Applying this rule to our case, yields the following constraint: $v\delta t = \Delta trace <= \frac{1}{2\omega}$. This equation characterizes the *temporal* sampling rate which is required to avoid temporal aliasing. In practice, video sequences of scenes with fast moving objects often contain temporal aliasing. We cannot control the frame rate $(\frac{1}{\delta t})$ nor object's motion (v). We can, however, decrease the spatial frequency upper bound ω by reducing the spatial resolution of each frame (i.e., apply a spatial low-pass-filter). This implies that for video sequences which inherently have high temporal aliasing, it may be necessary to compromise in spatial resolution of alignment in order to obtain correct temporal alignment. Therefore, the LPF (low pass filters) in our spatio-temporal pyramid construction (Sec. 3.2) should be adaptively selected in space and in time, in accordance with the rate of temporal changes. This method, however, is not applicable when the displacement of the moving object is larger than the object itself.

Appendix C: Empirical Evaluation

We quantitatively evaluated the accuracy of our direct sequence-to-sequence alignment algorithm on sequences where ground truth information was available. In the first experiment we warped a video sequence using known spatio-temporal parameters, to synthetically generate a second sequence. We then applied our method to the warped and the original sequences and compared the computed parameters with the known ones. This produced highly accurate results. The temporal error was less than 0.01 of a frame time, and spatial error was less than 0.02 pixel.

To generate a less synthetic example with ground truth, we split a video sequence into two sub-sequences – one containing the odd-fields, and one containing the even-fields. The two "field" sequences are related by a known temporal shift of 0.5 a frame time and a known spatial shift of a 0.5 pixel along the Y axis. Note, that in this case the data comes from the same camera, but from completely different sets of pixels (odd rows constitute one sequence and even rows constitute the other sequence). We repeated the experiment several (10) times using different sequences and different spatial models (affine, projective). In all cases the temporal error was smaller than 0.02 of a frame time (i.e., the recovered time shift between the two sequences was between 0.48 – 0.52). The error in the Y-shift was smaller than 0.03 pixel (i.e., the recovered Y-shift was between 0.47 – 0.53 pixel), and the overall error in spatial misalignment was less than 0.1 pixels.

To test a more realistic case of sequences obtained by two different cameras we performed the following experiment. Each of the two input sequences was split into two sub-sequences of odd and even fields, resulting in 4 sub-sequences: $Odd_1, Even_1, Odd_2, Even_2$. Because the ground truth is not known between the two sequences, it is therefore not known between $Odd_1 \leftrightarrow Odd_2$, $Odd_1 \leftrightarrow Even_2$, $Even_1 \leftrightarrow Odd_2$, $Even_1 \leftrightarrow Even_2$. However, what is known is how transformations of pairs of these sequences are related to each other. That is, if the time shift between Odd_1 and Odd_2 is Δt, then the time shift between $Even_1$ and $Even_2$ should be also Δt, and the time shift between Odd_1 and $Even_2$ should be $\Delta t + 0.5$. Similarly, a simple relation also holds for pairwise spatial transformations. This experiment was performed several times on several different sequences, and in all cases the temporal error was bounded by 0.05 frame time and the spatial error was bounded by 0.1 pixel.

Finally we verified the accuracy of alignment using three (or more) real video sequences: S_1, S_2, S_3. For each pair of sequences S_i and S_j, we computed the spatio-temporal misalignment between the sequences, denoted here by $\Delta(S_i \rightarrow S_j)$. The evaluation was based on the degree of transitivity, i.e., $\Delta(S_1 \rightarrow S_3)$ should be equal to $\Delta(S_1 \rightarrow S_2) + \Delta(S_2 \rightarrow S_3)$. Thus, we can use the following evaluation measure:

$$Err = ||\Delta(S_1 \rightarrow S_2) + \Delta(S_2 \rightarrow S_3) - \Delta(S_1 \rightarrow S_3)||.$$

This experiment was repeated several times, for several different sequences. The temporal error did not exceed 0.1 frame time, and was usually about 0.05 frame time. The spatial errors were on the order of 0.1 pixel.

Appendix D: Sequence Alignment as a Generalization of Image Alignment

We first show that the direct sequence-to-sequence alignment algorithm of Section 3.2 is a generalization of direct image-to-image alignment. When there are no temporal changes in the scene, and no camera motion, then $I(x,y) = S(x,y,t)$ where I is a single image in the sequence (i.e., all frames are equivalent), and the temporal derivatives within the sequence are zero: $S_t \equiv 0$. Therefore, the error function described in Eq. (5), reduces to:

$$\underbrace{ERR(\vec{P})}_{\text{seq-to-seq}} = \textstyle\sum_{x,y,t} S - S' + \begin{bmatrix} S'_x & S'_y & 0 \end{bmatrix} \begin{bmatrix} J_{spatial} & 0 \\ 0 & J_{temporal} \end{bmatrix} \begin{bmatrix} P_{spatial} \\ P_{temporal} \end{bmatrix} =$$

$$= \textstyle\sum_t \left(\sum_{x,y} I' - I + [I_x \; I_y] \, J_{spatial} \vec{P}_{spatial} \right) = \sum_t \underbrace{err(\vec{P}_{spatial})}_{\text{img-to-img}}$$

where $J_{spatial}$ is the $2 \times n$ "spatial minor" and $J_{temporal}$ is the $1 \times m$ "temporal minor", respectively, of the $3 \times (m+n)$ Jacobian matrix J (m,n are the number of temporal and spatial parameters of $\vec{P}$, respectively). This shows that in such cases the SSD function of Eq. (5) reduces to the image-to-image alignment objective function of [29], averaged over all frames[6].

The same holds for the feature-based sequence-to-sequence alignment algorithm (Section 3.1). When there are no changes in the sequences, feature points remain at the same image positions over time. Their trajectories thus become degenerate and reduce to points. Therefore, the feature-based sequence-to-sequence alignment algorithm reduces to a feature-based image-to-image algorithm with improved signal-to-noise ratio.

Namely, when there are no dynamic changes in the scene and no camera motion, sequence-to-sequence alignment may provide only improved signal-to-noise ratio, but no new information. However, when there are temporal changes over time, sequence-to-sequence alignment exploits more information than image-to-image alignment can. This is discussed at length in Section 4.

[6] A similar derivation for the error functions of [3, 18] is found in [8].

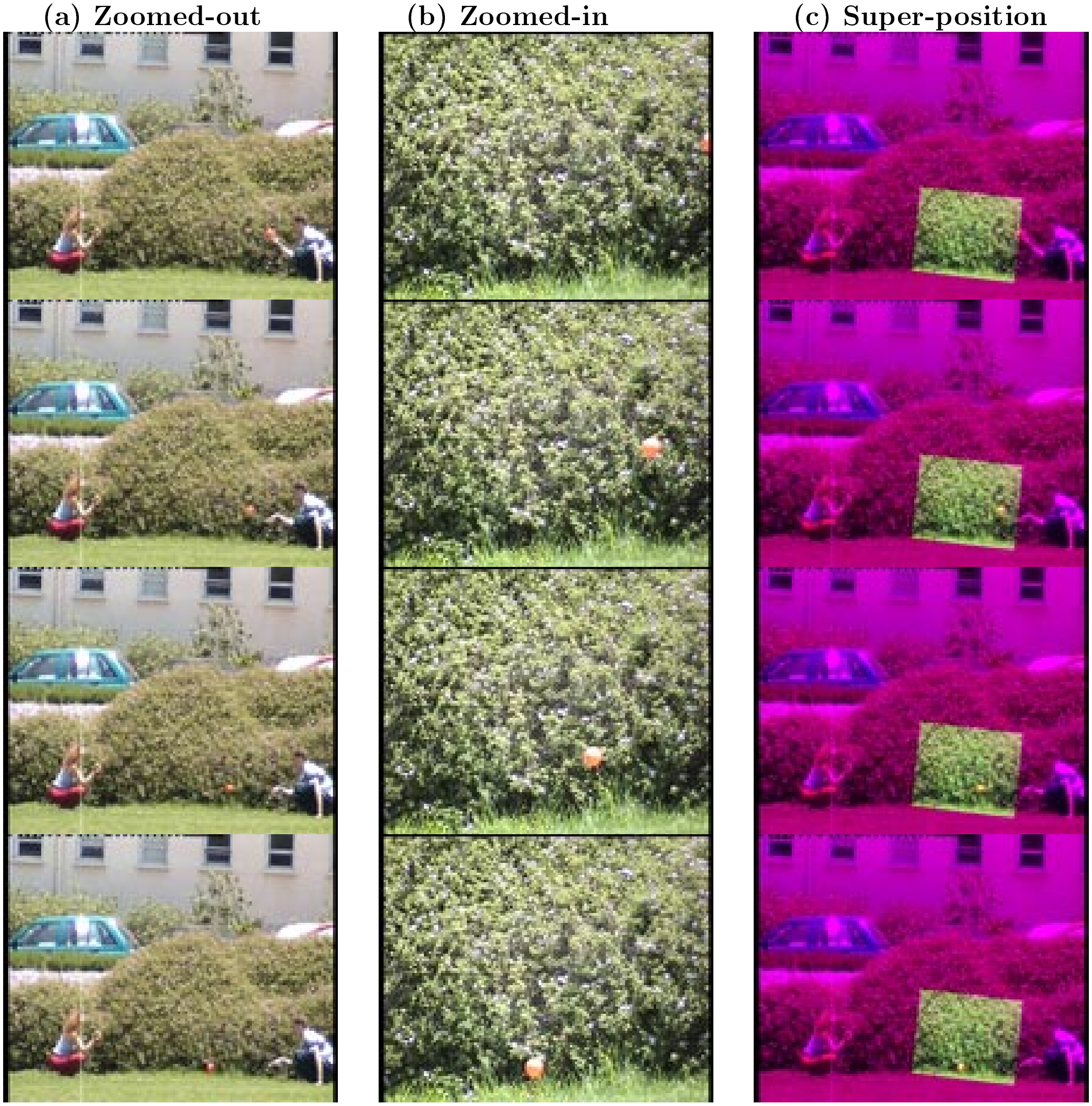


Figure 12: **Alignment of sequences obtained at different zooms.** *Columns (a) and (b) display four representative frames from the reference sequence and second sequence, showing a ball thrown from side to side. The sequence in column (a) was captured by a wide field-of-view camera, while the sequence in column (b) was captured by a narrow field-of-view camera (the ratio in zooms was approximately* 1 : 3*). The two sequences capture features at significantly different spatial resolution, which makes the problem of inter-camera image-to-image alignment very difficult. The dynamic information (the ball 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 feature-based algorithm of Section 3.1. The dark pink boundaries in (c) correspond to scene regions observed only by the reference (zoomed-out) camera.* **For full color sequences see www.wisdom.weizmann.ac.il/Seq2Seq.**