# מכון ויצמן למדע
# Weizmann Institute of Science

Thesis for the degree
**Doctor of Philosophy**

חבור לשם קבלת תואר
**דוקטור לפילוסופיה**

By
Yaron Caspi

מאת
ירון כספי

בנושא:

## התאמת סדרות של תמונות

# Sequence-to-Sequence Alignment

June 2002

תמוז תשס"ב

Submitted to the Scientific Council of
the Weizman Institute of Science
Rehovot, Israel

מוגש למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

# Abstract

*This thesis studies the problem of spatio-temporal alignment of video sequences, i.e., establishing correspondences in* time *and in* space *between two different video sequences of the same dynamic scene. It shows that temporal variations between image frames such as moving objects, changes in scene illumination, or camera ego-motion, are powerful cues for alignment. Such temporal variations cannot be exploited by standard image-to-image alignment techniques, as they are not captured by a single image, but only by a sequence of images. We show that by folding these new temporal cues and known spatial 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. This gives rise to a wide range of new video applications. These are discussed in this thesis.*

*The thesis investigates the cases where the sequences are recorded by uncalibrated video cameras with fixed internal and relative external parameters. However, the notion of sequence-to-sequence alignment/matching is more general, and is not restricted to those cases alone.*

# Acknowledgments

I would like to give special thanks to my advisor Dr. Michal Irani for her direct support and guidance. I greatly appreciate you for always challenging me and for setting the highest standards. I would have never achieved the same results without you.

I also wish to thank Lihi Zelkind, Ronen Basri and P. Anandan for many acadmic discussions and for strugling with some unreadble pre-versions of my papers. I (in the name of all other readers) thank you for making these papers easer to understand and more pleasant to read.

Many thanks to all the faculty members and administrative and technical support staff of the department of Computer Science at the Weizmann Institute of Science. You contributed greatly in generating, maintaining and supporting a lovely research environment necessary to conduct reasearch in genral and my study in particular. Specifically I would like to thank Professor Simon Ullman who urged me ten years ago not to compromise on the doctorate topic and to choose the subject that most interested me. Thank you, it did work out. I had three truly interesting and productive years. I also wish to share my deep admiration to the Fienberg Graduate School including teachers and students for making class both fascinating and fun. I would particularly like to mention the following students: Eli Shectman and Denis Simacov, the co-authors of two of my papers. I look forward to further fruitful collaboration.

Finally, I would like to thank my parents for always putting education at the top of the list and for teaching me the importance of persistence, and most of all to my dearest wife Yonit, who supported me day and night throughout these most wonderful three years of my life. Without your love and support I would have never been able to reach this final moment. Love and kisses.

Yaron

# Contents


**1 Introduction** **4**

1.1 What is sequence-to-sequence alignment? . . . . . 4

1.2 Exploiting Dynamic Cues (the motto of this thesis) . . . . . 4

1.2.1 Exploiting Scene Dynamics . . . . . 5

1.2.2 Exploiting Joint Motion of Cameras . . . . . 5

1.3 Sequence-to-Sequence Applications . . . . . 6

**2 Related Work** **7**

2.1 Image-to-Image Alignment . . . . . 7

2.2 Alignment of Sequences . . . . . 9

2.3 Action Recognition . . . . . 10

2.4 Integrating Visual Information . . . . . 11

2.4.1 Super-Resolution . . . . . 11

2.4.2 Sensor Fusion . . . . . 12

2.4.3 Panoramic (wide-screen) Movies . . . . . 12

2.5 Three Dimensional (Volumetric) Alignment . . . . . 12

**3 Summary** **13**

**4 List of Attached Papers** **13**

**5 References** **14**

**6 The Attached Papers** **20**

# 1 Introduction

This is a "paper-thesis", i.e., a thesis organized as a collection of papers. This document can be viewed as a road map of this paper-thesis. The attached papers, which form the core of this thesis, are listed in Chapter 3 (pp. 12). This document presents the general concept of the thesis and places the attached papers in the context within the overall thesis.

## 1.1 What is sequence-to-sequence alignment?

The problem of image-to-image alignment has been extensively studied over the past century. In using the term "image-to-image alignment", we refer to the problem of densely estimating point correspondences between two or more images. This thesis 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 $(x, y)$ in each frame (time) $t$ in one video sequence, find its corresponding frame $t'$ and pixel $(x', y')$ in another sequence, such that: $(x', y', t') = (x + u, y + v, t + w)$, where $(u, v, w)$ is the spatio-temporal displacement.

The typical scenario is when two cameras capture the same dynamic scene. Spatial misalignment results from the fact that the two cameras may be in different positions, have different orientations, and may also have different internal calibration parameters. These can be can be modeled by a $2D$ or $3D$ spatial transformations. Temporal misalignment results from the fact that the two cameras may not have been activated simultaneously, and possibly do not have the same frame rates (e.g., PAL and NTSC). This can be modeled by a $1D$ affine transformations in time.

There are two primary motivations for using sequence-to-sequence alignment: (1) It allows to resolve spatial ambiguities and to handle situations where image-to-image alignment fails. (2) Alignment and integration of information across multiple sequences both in space and in time gives rise to new video applications that are not possible when only spatial image alignment is used. These are shown in this thesis.

## 1.2 Exploiting Dynamic Cues (the motto of this thesis)

Image-to-image alignment methods (see Sec. 2.1 for several examples) are known to be accurate, and efficient. As implied by their name, image-to-image alignment methods are limited to the information contained in the images, i.e., the appearance (spatial information) of the images. This is not the case for video sequences. A video sequence is far more than a plain collection of images. It contains more information than any individual frame does. In particular, it captures information about the scene and camera dynamics. This additional information, denoted in this thesis as dynamic cues, forms an alternative/additional powerful cue for spatial (and temporal) alignment. These dynamic changes may result from either changes in the scene (e.g., moving

objects, changes in illumination), or changes in the cameras (e.g., the camera ego-motion). Such changes are not captured by any individual frame, but only by the entire sequence. It will be shown that both cases can be used to align sequences both in time and in space. Each of the two cases is discussed and illustrated in the sequel.

### 1.2.1 Exploiting Scene Dynamics

An example of the additional information in a video sequence is shown in Fig. 1. Alignment of image 1.a to image 1.b. is not uniquely defined when only information in individual images is used (see Fig. 1.c). However, a video sequence captures information about scene dynamics such as the trajectory of the moving object shown in Figs. 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).

Scene dynamics is not limited to moving objects. It also includes non-rigid changes in the scene (e.g., flowing water), changes in illumination, and more. All these changes are not captured by any of the individual frames, but are found *between* the frames. 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. Consequently, it forms an additional or sometimes an alternative powerful cue for alignment across sequences.

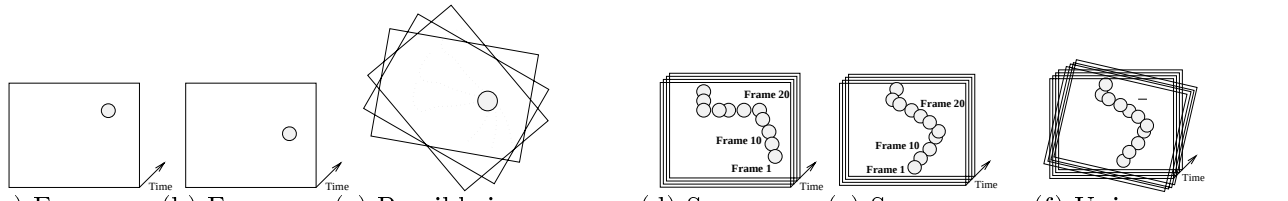
Information cues based on scene dynamics are studied in the first and second attached papers. The first (*"A Step Towards Sequence-to-Sequence Alignment"*) describes a method that exploits such cues for alignment directly from space-time brightness variations in the sequences. An alternative approach, where changes are tracked and the tracking results are aligned is illustrated and discussed in the second attached paper (*"Spatio-Temporal Alignment of Sequences"*).

While these two papers focus on recovery of 2D parametric transformations between the sequences, we have recently extended this idea to recovery of epipolar geometry between widely separated cameras as well. See the last attached paper (*"Feature-Based Sequence-to-Sequence Matching"*).

All these papers ((1),(2),(6)) describe algorithms that exploit the scene dynamics. In addition they also study the properties of the general concept of sequence-to-sequence alignment (beyond the specific proposed algorithms).

### 1.2.2 Exploiting Joint Motion of Cameras

Attached paper (3), *"Alignment of Non-Overlapping Sequences"* (and its extended journal version (4)) shows that "coherent appearance", which is the fundamental source of information in standard image alignment methods, can be replaced by "coherent temporal behavior" when matching image sequences. It shows that when two cameras are attached closely to each other (so that their centers of projection are very close), and move jointly in space, the induced frame-to-frame transformations within each sequence have correlated behavior across the two sequences. These




(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).*

papers show that this correlated behavior suffices to align two video sequences both in time and in space. Furthermore, the above observations are true even when the sequences have no spatial overlap. Thus, we are able to align non-overlapping sequences. This gives rise to a variety of applications discussed in Section 1.3.

## 1.3 Sequence-to-Sequence Applications

Attached paper (5) (*"Increasing Space-Time Resolution in Video"*) describes an application of sequence-to-sequence alignment. It shows how both the spatial resolution and the temporal resolution of a video camera can be exceeded by combining information from multiple video sequences. An increase in the temporal resolution is not possible when only image-to-image alignment is used.

The problem of image-based (i.e., spatial) super-resolution has been previously investigated by many researchers (see Section 2.4.1). In image-based super-resolution, 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 images. 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 super-resolution in time. By "temporal super-resolution" we refer to the recovery of rapid dynamic events that occur faster than regular frame-rate. Such dynamic events are not visible (or else observed incorrectly) in any of the input sequences, even if these are played in "slow-motion". Furthermore, attached paper (5) shows that we can combine spatial super-resolution with temporal super-resolution in a single framework.

Another family of applications that should be addressed by sequence-to-sequence alignment are applications that are particularly difficult for standard image alignment techniques. These include: (i) Alignment of sequences obtained at significantly different zooms (e.g., for surveillance applications), (ii) Alignment of multi-sensor sequences for multi-sensor fusion, and (iii) Recovery of large transformations and wide baseline matching.

Regular image alignment methods have difficulty in all the aforementioned cases, as features

which are visible to one camera may not even be observable by the other. This is true for different resolutions, different sensing modalities, and different viewpoints. Sequence-to-sequence alignment, however, is not as sensitive to these changes. The reason for this is that sequence-to-sequence alignment exploits dynamic information. Dynamic information is less effected by these (more invariant to) changes in imaging conditions across the two cameras. For example changes in sensing modalities affect the appearance, but do not affect the trajectory of motion induced by a moving object. This concept is illustrated in attached paper (6) (and (2)) for alignment that is based on common scene dynamics, and in attached papers (3,4) for alignment that is based on joint camera motion.

# 2 Related Work

## 2.1 Image-to-Image Alignment

The problem of image-to-image alignment has been studied extensively in the literature, and many different methods of alignment have been proposed. These approaches can be broadly classified into several categories: direct-based (or gradient-based), feature-based, region-based and statistical methods. Within each class, the particular methods differ in their final goal and in the minimization techniques used. Direct (gradient-based) methods are very popular for regular video applications in which the goal is generally to align frame $i$ to frame $i+1$. Horn and Schunck [34, 35, 36] and Lucas and Kanade [47] used this approach to compute optical flow. Bergen et. al. [7] described a formulation that can represent many of the 80's and early 90's alignment methods. Furthermore, using that formulation they derived many algorithms for a hierarchy of global motion models. Hanna [28] introduces a shape model into the alignment scheme. Several independent groups modeled the scene parallax (explicit representation of shape) using the "Plane+Parallax" model [45, 53], while others incorporated multi-frame approaches [38, 41, 78]. What enables us to classify all these methods as direct-based is that essentially all these approaches try to minimize the difference of intensity values:

$$||I(\vec{\mathbf{x}}) - I'(\vec{\mathbf{x}} + \vec{\mathbf{u}})||, \tag{1}$$

where $\vec{x}$ are image points, $||\cdot||$ represents some norm[1], and $\vec{u}$ represents the local displacement. The nature of the interrelationship between these $\vec{u}$s differs from method to method. Although these approaches are relatively old and well studied, new contributions continue to be made. See [1, 3, 61, 11] to list just a few.

Typical feature-based image alignment methods first apply a local operator to detect interest points in a pair of images (e.g., the Harris corner detector [30]). Once interest points are extracted

[1]This error measure was also substituted into monotonic function in order to incorporate robust statistics tools (e.g., [8, 9])

in the two images, robust estimation methods, such as RANSAC [21] and LMS [27], are used to find corresponding points and to extract the spatial transformation between the two images. The process is usually initialized by correlation-based matching (see for example [80, 77]).

In many cases feature-based approaches are presented in the context of recovery of the 3D structure of the scene (i.e., they are usually associated with stereo and structure from motion and not directly to the alignment problem. However, simple modifications (one might even claim simplifications) of these methods can transform them into standard alignment methods. A representative example of a feature-based alignment method is described by Hartley and Zisserman [31] (Chapter 3), followed by evaluation and error analysis (Chapter 4). Similarly, Stewart [65] illustrates that feature-based alignment is a special case of robust parameter estimation and illustrates its applicability to medical imaging [13, 14]. Thus, it could be argued that any algorithm for computing a 3D structure (e.g., [20]) can be adjusted to compute 2D parametric transformations.

Region-based methods are most commonly applicable to wide-base line alignment. The problem usually encountered by region-based methods is that both feature interest points and intensity values are sensitive to large changes in viewing conditions. Statistics of a large region are more likely to be characteristic properties that remain invariant under large changes of viewpoint. By "region-based" methods we refer to methods where the the aligned image portions are data dependent and are determined from the image content and not predefined (i.e., not blocks of size $n \times n$). Such methods match an image region/portion in one image to an image region in the second image using region characteristic (e.g., color texture), then uses these multiple region correspondences to recover the unknown alignment between the images. Pritchett and Zisserman [50] and Schaffalitzky and Zisserman [56] combined small textured region matches, locally aligned by affine transformations, into a more general model (e.g, homography). Basri and Jacobs [5, 6] search for a transformation that maps pixels of a given region from one image into the corresponding region in the other image. They showed that a small number of regions (3 for homography) suffices to uniquely determine the transformation parameters. Tao et. al. [68] exploit regions of consistent color, to match regions for stereo applications.

Similar to wide base-line problems, regular image-to-image alignment methods also encounter difficulties when trying to align two images captured by two sensors having different sensing modalities. Again, interest points and intensity values will perform poorly. Although the appearance in such cases may differ significantly, such images still share statistical properties. A popular approach, primarily for medical imaging, is alignment by maximizing the mutual information. This approach was first introduced by Viola and Wells [75], and has since been extended in several ways (see for example [32, 33]). Other statistical tools have also been proposed. Irani and Anandan [39] showed how to embed any local matching criteria into a global alignment framework. In particular they used normalized correlation surfaces. Fitzgibbon [22] modeled non-rigid scenes (such as flower fields) as an autoregressive process, where an optimal alignment is assumed to minimize the process noise component.

To summarize, image-to-image alignment methods assume that there is sufficient "similarity" between the two images, where the term "similarity" of images is used in the broadest sense to include all of the aforementioned methods. Consequently, image-to-image alignment methods implicitly share the basic assumption that there is sufficient overlap between the two images to allow extraction of common image properties. It is shown in attached papers (3) and (4) that this assumption is not required for some cases of sequences-to-sequence alignment (i.e., when the cameras move jointly).

## 2.2 Alignment of Sequences

In contrast to image alignment, where both the core alignment problem and the related applications (e.g., super-resolution, mosaicking, fusion, change detection, etc.) have been widely studied, only several studies have addressed the core problem of aligning sequences.

In their "Forest of Sensors" project [26], Grimson et al. suggested several applications of multiple collaborating sensors. See project website [63] for project summary and results. As part of that project Stein [64] and Lee et al. [46] developed a method for estimating a time shift and a homography between two sequences. The method is based on alignment of centroids of moving objects. However, there is a fundamental difference between [64, 46] and our approach. The centroids in [64, 46] were treated as an *unordered* collection of feature points and not as trajectories. Their features are based on temporal properties, but their alignment approach is based only on spatial properties of these features. In contrast, we enforce correspondences between space-time entities, in this case *trajectories* of moving objects. By doing so, we avoid the combinatorial complexity of establishing point matches of all points in all frames, resolve ambiguities in point correspondences, and allow for temporal correspondences at *sub-frame* accuracy. This is not possible when the points are treated independently (i.e., as a "cloud of points"), as in [64, 46].

A special case of this concept was provided by Wexler and Chellappa [76] who used a single moving object (a lamp in a dark room) captured by synchronized cameras to recover the cameras' pose and orientations and to calibrate the cameras.

The usefulness of simultaneously employing image constraints from multiple time instances was noted by several other researchers as well. Vadula et al. [73, 74], Zhang and Kambhamettu [79], and Tao et al. [69], all proposed combineing temporal alignment constraints and spatial alignment constraints into a single error function. The goal was usually to recover the scene structure, and they all used multiple synchronized cameras[2]. In practice, they all used two consecutive time instances and utilized the following observation: Assume that points $x(t)$ and $y(t)$ viewed by two different cameras are corresponding points at time instance $t$. If we can track (match) each of them independently, then the new pair of points $x(t+1)$ and $y(t+1)$ are also in correspondence. A different application that relies on this observation is described by Sawhney

[2]Tao et al. eventually only used the spatial constraints.

et al. [55]. Synchronized high-resolution and low-resolution cameras mounted on a stereo rig were used to construct a high-resolution stereo pair. They also used constraints from multiple consecutive time instances to recover accurate parallax at extremely high spatial resolution.

## 2.3 Action Recognition

The field of action recognition (including biological motion recognition, gesture recognition and gesture analysis) is closely related to sequence-to-sequence alignment. By action recognition, we refer here to studies that address the question of how to select the most relevant model stored in a predefined library with a given input sequence. More specifically: action recognition schemes often align the input sequence with all database sequences and choose the one that minimizes the alignment residual error. The main issue that must be considered is the source of misalignment between the video sequences. In sequence-to-sequence alignment (i.e., this thesis) we assume that the sequences capture the same dynamic scene using different (unsynchronized) cameras. In action recognition the sequences are captured at different times. In theory this problem should also encounter different viewpoints and different cameras in use. However, in many cases for the sake of simplicity, the database and input sequences are captured by the same camera. Also, in sequence-to-sequence alignment we can assume that the time scales are related by a global model (we used 1D affine transformation to compensate for different frame rates and different starting times) whereas in action recognition we can only assume monotonicity in time when correlating two actions.

Darrel and Pentland [18] have incorporated "dynamic time warping" taken from speech recognition to temporally match sequences of gestures. The temporal correspondence is recovered using dynamic programming, and the local score is based on correlation between images. The "invariance" to viewing direction is achieved by including many representative viewing directions. A view-invariant approach was introduced by Seitz and Dyer [58]. They used Tomasi and Kanade [71, 72] factorization-based rank constraint as their "alignment" criteria. Based on the sum of squares of singular values (except for the four largest ones), they temporally aligned different "cycles" (periodic motion with different time periods) of a repeated motion. A similar approach was proposed by Rao and Shah [51] (using a single centroid point and a rank-3 constraint) to recognize hand movements in a view-invariant manner.

Giese and Poggio [23, 24, 25] generalized the idea of representing a 3D object by linear combinations of three prototypical instances into modeling "biological motion patterns" using linear combinations of a few prototypical sequences. They reported that it is an "ill-posed" problem, as incorrect temporal shift can compensate for inaccurate spatial correspondence. A similar observation may be found in the first and second attached papers for some cases of sequence-to-sequence alignment.

Carlsson [17] represented consistency between sequences of walking people by a matrix of values. Each entry $(i, j)$ measures a rigidity constraint between few body locations in frame $i$ in one

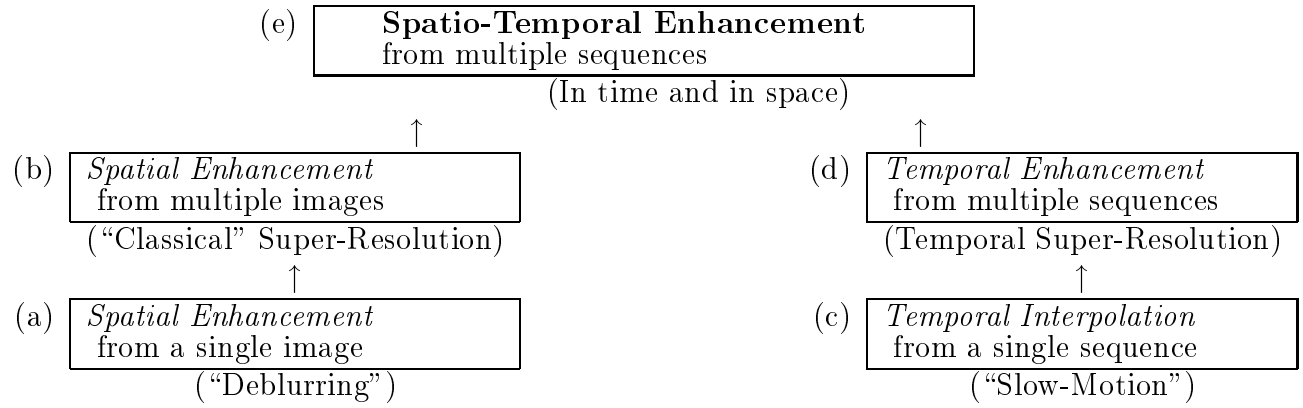



Figure 2: Hierarchy of enhancement problems.

sequence to the same body locations in frame $j$ in the other sequence. Now a dominant diagonal in this matrix represents correspondence between actions (walking people). Furthermore, tilted diagonals represents consistency (similar walking style) under different speed.

Mahmood et al. [66] proposed using the number of tracked feature points that obey the epipolar constraint $x(t)^T F x(t)$, where $F$ is the fundamental matrix, and $t$ is the time index, as the alignment criterion for recognizing action events. The construction of the fundamental matrix is based on corresponding feature points over several time instances. A generalization of this idea to extended space-time trajectories is provided in attached paper (6).

## 2.4 Integrating Visual Information

### 2.4.1 Super-Resolution

Fig. 2 displays a hierarchy of approach to image/sequence enhancement. A similar diagram can be found in Borman and Stevenson's [10] comprehensive review of super-resolution techniques. The hierarchy begins with single image enhancement Fig 2.(a). Block (b) in Fig 2 contains a wide variety of approaches that combine information from multiple images. From pioneering frequency-based approaches of Huang and Tsai [37], through back-projection approaches [42, 43, 44], Convex Sets (POCS) [49, 59, 62, 70] and Maximum A-Posteriori [16, 57] and many others ([2, 4, 15, 16, 19, 37, 43, 49, 60] to list but a few) all focus on spatial super-resolution. Elad [19] showed that these two approaches ( (a) and (b) ) are indeed interlinked. He described a unified formulation of image enhancement and image super-resolution.

A different segment of methods tries to increase the frame rate of video sequences (Fig 2.(c)). Techniques for increasing the frame rate of a single sequence (slow-motion) usually perform time-interpolation, either by direct interpolation of intensity values (sometimes known as "tweening") or by tracking moving objects. Such applications can be found in commercial products, e.g.,

$Retimer^{TM}$ by RealVis [52]. However, they do not perform temporal super-resolution (Fig 2.(d)). By "temporal super-resolution" we mean recovery of rapid dynamic events that occur faster than regular frame rate. Such dynamic events are not visible (or else observed incorrectly) in any of the input sequences, even if these are played in "slow-motion".

In contrast to the vast body of research on integrating information from different images to produce higher quality output images, we did not find methods which combine multiple sequences to address the equivalent task in time (Fig. 2.(d)) or both in time and in space (Fig. 2.(e)). On the top of their two branches, Borman and Stevenson place "spatio-temporal resolution enhancement". However, no concrete method has ever been presented. The space-time super-resolution method presented in this thesis belongs to Fig. 2.(d)) and 2.(e)).

### 2.4.2 Sensor Fusion

The previous section illustrated that alignment can be used to exceed the limited resolution of a single video camera. Another limited property of sensors is their effective wave length. Visible light cameras capture light with wavelengths of between $0.3\mu m - 0.8\mu m$, while Infra-Red cameras are sensitive to radiation of wave length between $3\mu m - 5\mu m$, (or else $8\mu m - 12\mu m$). Applications that combine different sensing modalities are usually denoted by "image fusion". Kolczynski and Burt [12] proposed a method for image fusion. Their method assumes that the multi-sensor images have been pre-aligned. A similar application for fusion of sequences having multiple sensing modalities is described in attached papers (2)-(4).

### 2.4.3 Panoramic (wide-screen) Movies

Mosaics [40, 13, 15, 29, 54, 67, 83] is another widely used application that relies on accurate image alignment. In this case, the limited field of view is the physical property that is being extended. Attached papers 3 and 4 extend this application to dynamic scenes and provide means for generation of wide-field of view movies.

## 2.5 Three Dimensional (Volumetric) Alignment

Another research domain that is related to sequence-to-sequence is volumetric alignment, typically used in $3D$ imaging systems (e.g., medical imaging systems). In these cases the goal is to align "voxels" having $X, Y, Z$ coordinates, while aligning sequence elements have $X, Y, T$ coordinates. There is a fundamental difference between the two domains. In the case of sequence-to-sequence alignment, the spatial and temporal dimensions are very different in nature and may not be intermixed. Applying regular 3D volumetric transformations (e.g., a 3D affine transformation) to sequences may mix between time ($t$) and space ($x, y$), and is therefore not applicable. Sequence-to-sequence transformations must be separable in time ($t$) and space ($x, y$). A few examples of methods for $3D$ volumetric ($x, y, z$) alignment and integration are listed next:

Hermosillo and Faugeras [32, 33] aligned MRI images. Oz et al. [48] applied super-resolution algorithms to MRI data. In this example the several diffusion-weighted imaging single-shots shifted at sub-pixel accuracy are combined to construct a single 3D high-resolution MRI image. Note, that in this example shifts along the three axes (X-Y shifts and the Z shifts) are due to different physical origins. Three-dimensional affine transformation were used by Zhou et al. [81, 82] to estimate cloud (hurricane) motion and structure.

# 3 Summary

This thesis presents some approaches for establishing correspondences in *time* and in *space* between two different video sequences of the same dynamic scene, and its applications. It shows that temporal variations between image frames such as moving objects, changes in scene illumination, or the camera ego-motion, are powerful cues for alignment. We show that by folding these new temporal cues together with commonly used spatial 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. We further show that this gives rise to new video applications that are not possible when only image-to-image is used (such as temporal super-resolution or alignment of non-overlapping sequences).

This thesis focuses on a particular task - alignment of sequences. However, by illustrating the contribution of temporal cues for solving the alignment problem, it argues in favor of exploiting temporal cues when addressing other vision tasks as well

# 4 List of Attached Papers

1. Y. Caspi and M. Irani, "A Step Towards Sequence-to-Sequence Alignment," in *proceedings IEEE Conference on Computer Vision and Pattern Recognition*, II:682–689, Hilton Head Island, South Carolina, June 2000.

2. Y. Caspi and M. Irani, "Spatio-Temporal Alignment of Sequences," *to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence.*

3. Y. Caspi and M. Irani, "Alignment of Non-Overlapping Sequences," *proceedings IEEE International Conference on Computer Vision*, II:76–83, Vancouver Canada July 2001.

4. Y. Caspi and M. Irani, "Aligning Non-Overlapping Sequences," *to appear in International Journal of Computer Vision.*

5. E. Shechtman, Y. Caspi, and M. Irani, "Increasing Space-Time Resolution in Video," in *European Conference on Computer Vision, I:753–768, Copenhagen, Denmark, May 2002.*

6. Y. Caspi, D. Simakov, and M. Irani, "Feature-Based Sequence-to-Sequence Matching," in *Vision and Modelling of Dynamic Scenes Workshop, Copenhagen, Denmark, June, 2002.*

# 5 References

# 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 T. Kanade. Limits on super-resolution and how to break them. In *Computer Vision and Pattern Recognition (CVPR)*, Hilton Head Island, South Carolina, June 2000.

[3] 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.

[4] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *European Conference on Computer Vision (ECCV)*, pages 312–320, Cambridge, UK, April 1996.

[5] R. Basri and D. Jacobs. Recognition using region correspondences. *International Journal of Computer Vision*, 25(2):141–162,, 1997.

[6] R. Basri and D. Jacobs. Projective alignment with regions. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(5):519–527, 2001.

[7] 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.

[8] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *International Conference on Computer Vision (ICCV)*, pages 231–236, Berlin, Germany, May 1993.

[9] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding (CVIU)*, 63(1):75–104, Jan 1996.

[10] S. Borman and R. Stevenson. Spatial resolution enhancement of low-resolution image sequences - a comprehensive review with directions for future research. Technical report, Laboratory for Image and Signal Analysis (LISA), University of Notre Dame, Notre Dame, July 1998.

[11] P. Bride and P. Meer. Global registration via direct methods: a statistical approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, Hawaii, Dec. 2001.

[12] P.R. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *International Conference on Computer Vision (ICCV)*, pages 173–182, Berlin, Germany, May 1993.

[13] A. Can, C. V. Stewart, and B. Roysam. Robust hierarchical algorithm for constructing a mosaic from images of the curved human retina. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 286–292, Ft. Collins, Colorado, June 1999.

[14] A. Can, C.V. Stewart, B. Roysam, and H.L. Tanenbaum. A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: Application to mosaicking the curved human retina. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II:585–591, Hilton Head Island, South Carolina, 2000.

[15] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 885–891, June 1998.

[16] D. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *International Conference on Pattern Recognition (ICPR)*, pages 600–605, Barcelona, Spain, September 2000.

[17] S. Carlsson. Recognizing walking people. In *European Conference on Computer Vision (ECCV)*, pages 1842–1843, Dublin, Ireland, June 2000.

[18] T. Darrell and A. Pentland. Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, June 1993. IEEE Computer Society Press.

[19] M. Elad. Super-resolution reconstruction of images. Ph.D. Thesis, Technion Israel Institute of Technology, December 1996.

[20] O. Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1996.

[21] 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.

[22] A. W. Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 662–670, July 2001.

[23] 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.

[24] M. A. Giese and T. Poggio. Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. In *IEEE Workshop on Multi-View Modeling and Analysis of Visual Scene*, pages 73–80, Fort Collins, Colorado, June 1999.

[25] 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.

[26] E. Grimson, P. Viola, O. Faugeras, T. Lozano-Perez, T. Poggio, and S. Teller. A forest of sensors. In *DARPA97*, pages 45–51, 1997.

[27] 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.

[28] K. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *IEEE Workshop on Visual Motion*, pages 156–162, Princeton, New Jersey, October 1991.

[29] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *Proc. of the Workshop on Applications of Computer Vision II*, Sarasota, Florida, 1994.

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

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

[32] C. Chefd'Hotel G. Hermosillo and O. Faugeras. A variational approach to multi-modal image matching. In *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, University of British Columbia, Vancouver, Canada, July 2001.

[33] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. A variational approach to multi-modal image matching. Technical Report 4117, INRIA, February 2001.

[34] B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.

[35] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[36] B.K.P. Horn and B.G. Schunck. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, June 1988.

[37] T.S. Huang and R.Y. Tsai. Multi-frame image restoration and registration. In T.S. Huang, editor, *Advances in Computer Vision and Image Processing*, volume 1, pages 317–339. JAI Press Inc., 1984.

[38] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *International Conference on Computer Vision (ICCV)*, Corfu, September 1999.

[39] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *International Conference on Computer Vision (ICCV)*, pages 959–966, India, January 1998.

[40] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. In *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application*, volume 8, pages 327–351, May 1996.

[41] M. Irani, P. Anandan, and Meir Cohen. Direct recovery of planar-parallax from multiple frames. In *ICCV'99 Workshop:Vision Algorithms 99*, Corfu, September 1999.

[42] M. Irani and S. Peleg. Super resolution from image sequences. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 115–120, Atlantic City, New Jersey, June 1990.

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

[44] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion and transparency. In *Journal of Visual Communication and Image Representation*, volume 4, pages 324–335, December 1993.

[45] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: parallax based approach. In *International Conference on Pattern Recognition (ICPR)*, pages 685–688, 1994.

[46] 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.

[47] 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.

[48] G. Oz, H. Greenspan, N. Kiryati, and S. Peled. MRI inter-slice reconstruction using super-resolution. In *Medical Image Computing and Computer-Assisted Intervention*, October 2001.

[49] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. In *IEEE Trans. on Image Processing*, volume 6, pages 1064–1076, August 1997.

[50] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *International Conference on Computer Vision (ICCV)*, pages 754–760, India, January 1998.

[51] Cen Rao and Mubarak Shah. View invariance in action recognition. In *CVPR*, pages 11–13, Kauai, Hawaii, Dec 2001.

[52] RealVis. Retimer. *http://www.realviz.com*, 2000.

[53] H. Sawhney. 3D geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 929–934, Seattle, Washington, June 1994.

[54] 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.

[55] H. S. Sawhney, Y. Guo, K. Hanna, R. Kumar, S. Adkins, and S. Zhou. Hybrid stereo camera: an IBR approach for synthesis of very high resolution stereoscopic image sequences. In *SIGGRAPH*, volume 28, Los Angeles, California, Aug. 2001.

[56] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *International Conference on Computer Vision (ICCV)*, July 2001.

[57] R. S. Schultz and R. L. Stevenson. Extraction of high-resolution frames from video sequences. In *IEEE Trans. Image Processing*, volume 5, pages 996–1011, June 1996.

[58] S. M. Seitz and C. R. Dyer. View-invariant analysis of cyclic motion. In *International Journal of Computer Vision*, volume 25(3), 1997.

[59] M. I. Sezan. An overview of convex projections theory and its application to image recovery problems. In *Ultramicroscopy*, volume 40, pages 55–67, 1992.

[60] J. Shin, J. Paik, J. R. Price, and M.A. Abidi. Adaptive regularized image interpolation using data fusion and steerable constraints. In *SPIE Visual Communications and Image Processing*, volume 4310, January 2001.

[61] H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *IJCV*, 36(2):101–130, February 2000.

[62] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. In *Journal of the Optical Society of America A*, volume 6 (11), pages 1715–1726, 1989.

[63] C. Stauffer. A forest of sensors overview. *http://www.ai.mit.edu/projects/vsam*, 1998.

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

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

[66] 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.

[67] 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.

[68] H. Tao and H. S. Sawhney. Global matching criterion and color segmentation based stereo. In *in Proc. Workshop on the Application of Computer Vision*, pages 246–253, December 2000.

[69] H. Tao, H. S. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. In *CVPR*, Kauai, Hawaii, Dec. 2001.

[70] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan. High-resolution image reconstruction from lower-resolution image sequences and space varying image restoration. In *International Conference on Acoustics Speech and Signal Processing*, volume III, pages 169–172, San Francisco, California, March 1992.

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

[72] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, November 1992.

[73] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the 7th International Conference on Computer Vision*, volume 2, pages 722 – 729, September 1999.

[74] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6d. In *Computer Vision and Pattern Recognition (CVPR)*, Hilton Head Island, South Carolina, June 2000.

[75] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *International Conference on Computer Vision (ICCV)*, pages 16–23, 1995.

[76] Y. Wexler and R. Chellappa. View synthesis using convex and visual hulls. In *British Machine Vision Conference*, Manchester, U.K, 2001.

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

[78] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(10):1105–1116, October 2000.

[79] Y. Zhang and C. Kambhamettu. Scene flow and structure recovery from multiview image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 674–681, Hilton Head Island, South Carolina, June 2000.

[80] 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.

[81] L. Zhou, C. Kambhamettu, and D. Goldgof. Extracting nonrigid motion and 3d structure of hurricanes from satellite image sequences without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 280–285, Fort Collins, CO., June 1999.

[82] L. Zhou, C. Kambhamettu, and D. Goldgof. Fluid structure and motion analysis from multispectrum 2d cloud image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 744–751, Hilton Head Island, South Carolina, June 2000.

[83] A. Zomet and S. Peleg. Efficient super-resolution and applications to mosaics. In *International Conference on Pattern Recognition (ICPR)*, Barcelona, Spain, September 2000.

# 6 The Attached Papers

# A Step Towards Sequence-to-Sequence Alignment

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

## Abstract

This paper presents an approach for establishing correspondences in *time* and in *space* between two different video sequences of the same dynamic scene, recorded by stationary uncalibrated video cameras. The method *simultaneously* estimates both *spatial alignment* as well as *temporal synchronization* (temporal alignment) between the two sequences, using all available spatio-temporal information. 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.

We also present a *"direct"* method for sequence-to-sequence alignment. The algorithm simultaneously estimates spatial and temporal alignment parameters directly from *measurable sequence quantities*, without requiring prior estimation of point correspondences, frame correspondences, or moving object detection. Results are shown on real image sequences taken by multiple video cameras.

## 1 Introduction

The problem of image-to-image alignment has been extensively studied in the literature. By *"image-to-image alignment"* we refer to the problem of densely estimating point correspondences between two or more images (either taken by a single moving camera, or by multiple cameras), i.e., for each pixel $(x, y)$ in one image, find its corresponding pixel in the other image: $(x', y') = (x+u, y+v)$, where $(u, v)$ is the spatial displacement. This paper addresses a different problem – the problem of *"sequence-to-sequence alignment"*, which establish correspondences both in *time* and in *space* between multiple *sequences* (as opposed to multiple images). Namely, for each pixel $(x, y)$ in each frame (time) $t$ in one sequence, find its corresponding frame $t'$ and pixel $(x', y')$ in the other sequence: $(x', y', t') = (x+u, y+v, t+w)$, where $(u, v, w)$ is the *spatio-temporal* displacement.

The need for sequence-to-sequence alignment exists in many real-world scenarios, where multiple video cameras record information about the same scene over a period of time. Some examples are: News items commonly documented by several media crews; sports events covered by at least a dozen cameras recording the same scene from different view points; wide-area surveillance of the same scene by multiple cameras from different observation points. Grimson-et-al [7] suggested a few applications of multiple collaborating sensors. Reid and Zisserman [5] combined information from two independent sequences taken at the $66^{th}$ World Cup, to resolve the controversy regarding the famous goal. They *manually* synchronized the sequences, and then computed spatial alignment between selected corresponding images (i.e., image-to-image alignment). This is an example where spatio-temporal sequence-to-sequence alignment may provide enhanced alignment.

Image-to-image alignment methods are *inherently* restricted to the information contained in individual images – the spatial variations *within* an image (which corresponds to scene appearance). However, a video sequence contains much more information than any individual frame does. Scene dynamics (such as moving object, changes in illumination, etc) is a property that is inherent to the *scene*, and is thus common to all sequences taken from different video cameras. It therefore forms an *additional* powerful cue for alignment.

Stein [6] proposed an elegant approach to estimating spatio-temporal correspondences between two sequences based on alignment of *trajectories of moving objects*. Centroids of moving objects were detected and tracked in each sequence. Spatio-temporal alignment parameters were then seeked, which would bring the trajectories in the two sequences into alignment. No static-background information was used in this step[1]. This approach is hence referred to in our paper as *"trajectory-to-trajectory alignment"*. Giese and Poggio [3] also used trajectory-to-trajectory alignment

[1] In a later step [6] refines the spatial alignment using static background information. However, the temporal alignment is already fixed at that point.

to classify human motion patterns. Both [6, 3] reported that using temporal information (i.e., the trajectories) alone for alignment across the sequences may not suffice, and can often lead to inherent ambiguities between temporal and spatial alignment parameters.

This paper proposes an approach to *sequence-to-sequence alignment*, which simultaneously uses all available spatial and temporal information within a sequence. We show that when there is no temporal information present in the sequence, our approach reduces to image-to-image alignment. However, when such information exists, it takes advantage of it. Similarly, we show that when no static spatial information is present, our approach reduces to trajectory-to-trajectory alignment. Here too, when such information is available, it takes advantage of it. Thus our approach to sequence-to-sequence alignment combines the benefits of image-to-image alignment with the benefits of trajectory-to-trajectory alignment, and is a generalization of both approaches. We show that it resolves many of the inherent ambiguities associated with each of these two classes of methods.

We also present a specific algorithm for sequence-to-sequence alignment, which is a generalization of the direct image alignment method of [1]. It is currently assumed that the sequences are taken by stationary video cameras, with fixed (but *unknown*) internal and external parameters. Our algorithm simultaneously estimates spatial and temporal alignment parameters *without* requiring prior estimation of point correspondences, frame correspondences, moving object detection, or detection of illumination variations.

The remainder of this paper is organized as follows: Section 2 presents our direct method for the spatio-temporal sequence-to-sequence alignment algorithm. Section 3 studies some inherent properties of sequence-to-sequence alignment, and compares it against image-to-image alignment and trajectory-to-trajectory alignment. Section 4 provides selected experimental results on real image sequences taken by multiple unsynchronized and uncalibrated video cameras. Section 5 concludes the paper.

## 2 The Sequence Alignment Algorithm

The scenario addressed in this paper is when the video cameras are stationary, with fixed (but *unknown*) internal and external parameters. The recorded scene can change dynamically, i.e., it can include multiple independently moving objects (there is no limitation on the number of moving objects or their motions), it can include changes in illumination over time (i.e., within the sequence), and/or other temporal changes. *Temporal misalignment* can result from the fact that the two input sequences can be at different frame rates (e.g., PAL and NTSC), or may have a time-shift (offset) between them (e.g., if the cameras were not activated simultaneously). The temporal shift may be at sub-frame units. These factors give rise to a 1-D affine transformation in time. *Spatial misalignment* results from the fact that the two cameras are in different positions and have different internal calibration parameters. The spatial alignment can range from 2D parametric transformations to more general 3D transformations.

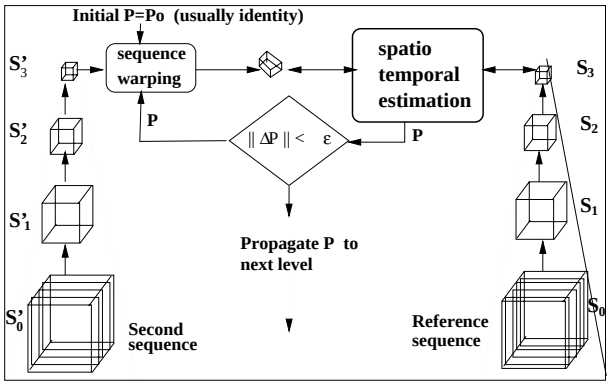



**Figure 1. The hierarchical spatio-temporal alignment framework** *A volumetric pyramid is constructed for each input sequence, one for the reference sequence (on the right side), and one for the second sequence (on the left side). The spatio-temporal alignment estimator is applied iteratively at each level. It refines the approximation based on the residual misalignment between the reference volume and warped version of the second volume (drawn as a skewed cube). The output of current level is propagated to the next level to be used as an initial estimate.*

This section presents an algorithm for sequence to sequence alignment. The algorithm is a generalization of the hierarchical direct image-to-image alignment method of Bergen-et-al [1], and Irani-et-al [4]. While this specific algorithm is a direct brightness-based method, the concept of sequence-to-sequence alignment presented in this paper is more general, and can similarly be used to extend feature-based image-to-image alignment methods as well.

In [1, 4] the spatial alignment parameters were recovered directly from image brightness variations, and the coarse-to-fine estimation was done using a Gaussian image pyramid. This is generalized here to recover the *spatial* and *temporal* alignment parameters directly from sequence brightness variations, and the coarse-to-fine estimation is done within a *volumetric sequence pyramid*. An image sequence is handled as a *volume* of three dimensional data, and not as a set of two-dimensional images. Pixels become spatio-temporal "voxels" with three coordinates: $(x, y, t)$, where $x, y$ denote spatial image coordinates, and $t$ denotes time. The multi-scale analysis is done both in *space* and in *time*.

Fig 1 illustrates the hierarchical spatio-temporal estima-

tion framework. The rest of this section is organized as follows: Section 2.1 describes the core step (the inner-loop) within the iterate-refine algorithm. In particular, it generalizes the image brightness constraint to handle sequences. Section 2.2 presents a few sequence-to-sequence alignment models which were implemented in the current algorithm. Section 2.3 presents the volumetric sequence-pyramid. Section 2.4 summarizes the algorithm.

### 2.1 The Sequence Brightness Error

Let $S, S'$ be two input image sequences, where $S$ denotes the reference sequence, $S'$ denotes the second sequence. Let $(x, y, t)$ be a spatio-temporal "voxel" in the reference sequence $S$. Let u,v be its spatial displacements, and w be its temporal displacement. Denote by $\vec{P} = (\vec{P}_{spatial}, \vec{P}_{temporal})$ the unknown alignment parameter vector. While every "voxel" (x,y,t) has a different local spatio-temporal displacement (u,v,w), they are all *globally* constrained by the parametric model $\vec{P}$. Therefore, every "voxel" (x,y,t) provides one constraint on the global parameters. A global constraint on $\vec{P}$ is obtained by minimizing the following SSD objective function:

$$ERR(\vec{P}) = \sum_{x,y,t} (S'(x,y,t) - S(x-u, y-v, t-w))^2, \quad (1)$$

where: $u = u(x, y, t; \vec{P})$, $v = v(x, y, t; \vec{P})$, $w = w(x, y, t; \vec{P})$. $\vec{P}$ is estimated using the Gauss-Newton minimization technique. This is done by linearizing the difference term $(S' - S)$ in Eq. (1). This step results in a new error term, which is quadratic in the unknown displacments (u,v,w):

$$ERR(\vec{P}) = \sum_{x,y,t} (e(x, y, t; \vec{P}))^2, \quad (2)$$

where,

$$e(x, y, t; \vec{P}) = S'(x, y, t) - S(x, y, t) + [u\ v\ w] \nabla S(x, y, t), \quad (3)$$

and $\nabla S = [S_x S_y S_t] = [\frac{\partial S}{\partial x}\ \frac{\partial S}{\partial y}\ \frac{\partial S}{\partial t}]$ denotes a spatio-temporal gradient of the sequence S. Eq. (3) directly relates the unknown displacements $(u, v, w)$ to measurable brightness variations within the sequence. To allow for large spatio-temporal displacements $(u, v, w)$, the minimization of Eq. (1) is done within an iterative-warp coarse-to-fine framework (see Sections 2.3 and 2.4).

Note that the objective function in Eq. (2) integrates all available spatio-temporal information in the sequence. Each spatio-temporal "voxel" (x,y,t) contributes as much information as it reliably can to each unknown . For example, a "voxel" which lies on a *stationary vertical edge*, (i.e., $S_x \neq 0, S_y = S_t = 0$), affects only the estimation of the parameters involved in the horizontal displacement $u(x, y, t; \vec{P})$. Similarly, a "voxel" in a uniform region ($S_x = S_y = 0$) which undergoes a temporal change ($S_t \neq 0$), e.g., due to variation in illumination, contributes only to the estimation of the parameters affecting the *temporal* displacement $w(x, y, t; \vec{P})$. A highly textured "voxel" on a moving object (i.e., $S_x \neq 0, S_y \neq 0, S_t \neq 0$), contributes to the estimation of *all* the parameters.

### 2.2 Spatio-Temporal Alignment Models

In our current implementation, $\vec{P} = (\vec{P}_{spatial}, \vec{P}_{temporal})$ was chosen to be a parametric transformation. Let $\vec{p} = (x, y, 1)^T$ denote the homogeneous *spatial* coordinates of a spatio-temporal "voxel" $(x, y, t)$. Let $H$ be the $3 \times 3$ matrix of the *spatial* parametric transformation between the two sequences. Denoting the rows of $H$ by $[H_1, H_2, H_3]^T$, the spatial displacement can be written as: $u(x, y, t) = \frac{H_1\vec{p}}{H_3\vec{p}} - x$, and $v(x, y, t) = \frac{H_2\vec{p}}{H_3\vec{p}} - y$. Note that $H$ is common to all frames, because the cameras are stationary. When the two cameras have different frame rates (such as with NTSC and PAL) and possibly a time shift, a 1-D affine transformation suffices to model the temporal misalignment between the two sequences: $w(t) = d_1 t + d_2$ (where $d_1$ and $d_2$ are real numbers). We have currently implemented two different spatio-temporal parametric alignment models:

**Model 1:** *2D spatial affine transformation & 1D temporal affine transformation.* The spatial 2D affine model is obtained by setting the third row of $H$ to be: $H_3 = [0, 0, 1]$. Therefore, for 2D spatial affine and 1D temporal affine transformations, the unknown parameters are: $\vec{P} = [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ d_1\ d_2]$, i.e., eight unknowns. The individual voxel error of Eq. (3) becomes: $e(x, y, t; \vec{P}) = S' - S + [(H_1\vec{p} - x)\ (H_2\vec{p} - y)\ (d_1 t + d_2)]\ \nabla S$, which is linear in all unknown parameters.

**Model 2:** *2D spatial projective transformation & a temporal offset.* In this case, $w(t) = d$ ($d$ is a real number, i.e., could be a sub-frame shift), and $\vec{P} = [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ h_{31}\ h_{32}\ h_{33}\ d]$. Each spatio-temporal "voxel" (x,y,t) provides one constraint:

$$e(x, y, t; \vec{P}) = S' - S + \left[(\frac{H_1\vec{p}}{H_3\vec{p}} - x)\ (\frac{H_2\vec{p}}{H_3\vec{p}} - y)\ d\right] \nabla S. \quad (4)$$

The 2D projective transformation is not linear in the unknown parameters, and hence requires some additional manipulation. To overcome this non-linearity, Eq. (4) is multiplied by the denominator $(H_3\vec{p})$, and renormalized with its current estimate from the last iteration, leading to a slightly different error term:

$$e_{new}(x, y, t; \vec{P}) = H_3\vec{p} / \hat{H}_3\vec{p} \cdot e_{old}(x, y, t; \vec{P}), \quad (5)$$

where $\hat{H}_3$ is the current estimate of $H_3$ in the iterative process, and $e_{old}$ is as defined in Eq. (4). Let $\hat{H}$ and $\hat{d}$ be the current estimates of $H$ and $d$, respectively. Substituting $H =$

$\hat{H} + \delta H$ and $d = \hat{d} + \delta d$ into Eq. (5), and neglecting high-order terms, leads to a new error term, which is linear in all unknown parameters ($\delta H$ and $\delta d$). We found in our experiments that in addition to second order terms (e.g, $\delta H \delta d$), the first order term $\hat{d}\delta H_3$ is also negligible and can be ignored.

In the above implementations $\vec{P}$ was assumed to be a parametric transformation. However, the presented framework is more general, and is not restricted to parametric transformations alone. $(u, v, w)$ can be equally expressed in terms of 3D parameters (the epipole, the homography, and the shape). See [1] for a hierarchy of possible spatial alignment models.

### 2.3 Spatio-Temporal Volumetric Pyramid

The estimation step described in section 2.1 is embedded in an iterative-warp coarse-to-fine estimation framework. This is implemented within a spatio-temporal volumetric pyramid. Multi-scale analysis provides three main benefits: (i) Larger misalignments can be handled, (ii) the convergence rate is faster, and (iii) it avoids getting trapped in local minima. These three benefits are discussed in [1] for the case of spatial (image) alignment. Here they are extended to the temporal domain as well.

The Gaussian[2] image pyramid [2] is generalized to a Gaussian sequence (volumetric) pyramid. The highest resolution level is defined as the input sequence. Consecutive lower resolution levels are obtained by low-pass filtering (LPF) both in *space* and *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 $256 \times 256$ pixels, then the next resolution level contains a sequence of 32 frames of size $128 \times 128$, etc. A discussion of the trade-offs between spatial and temporal low-pass-filtering may be found in Appendix A.

### 2.4 Summary of the Algorithm

The iterative-warp coarse-to-fine estimation process is schematically described in Fig 1, and is summarized below:
1. Construct two spatio-temporal volumetric pyramids, one for each input sequence: $(S_0 := S), S_1, S_2..S_L$ and $(S'_0 := S'), S'_1, S'_2..S'_L$. Set $\vec{P} := \vec{P}_0$ (usually the identity transformation).
2. For every resolution level, $l = L..0$, do:
   (a) Warp $S'_l$ using the current parameter estimate: $\hat{S}'_l := warp(S'_l; \vec{P})$.
   (b) Refine $\vec{P}$ according to the residual misalignment between the reference $S_l$ and the warped $\hat{S}'_l$ (see Section 2.1).
   (c) Repeat steps (a) and (b) until $||\Delta P|| < \epsilon$.
(3) Propagate $\vec{P}$ to the next pyramid level $l - 1$, and repeat the steps (a),(b),(c) for $S_{l-1}$ and $S'_{l-1}$.

The resulting $\vec{P}$ is the spatio-temporal transformation, and the resulting alignment is at sub-pixel spatial accuracy, and sub-frame temporal accuracy. Results of applying this algorithm to real image sequences are shown in Section 4.

[2] A Laplacian pyramid can equally be used.

## 3 Properties of Sequence-to-Sequence Alignment

This section studies several inherent properties of sequence-to-sequence alignment. In particular it is shown that sequence-to-sequence alignment is a generalization of image-to-image alignment and of trajectory-to-trajectory alignment approaches. It is shown how ambiguities in spatial alignment can often be resolved by adding temporal cues, and vice versa, how temporal ambiguities (reported in [6, 3]) can be resolved by adding spatial cues. These issues are discussed in Sections 3.1 and 3.2. We further show that temporal information is not restricted to moving objects. Different types of temporal events, such as changes in scene illumination, can contribute useful cues (Section 3.3). These properties are illustrated by examples from the algorithm presented in Section 2. However, the properties are general, and are not limited to that particular algorithm.

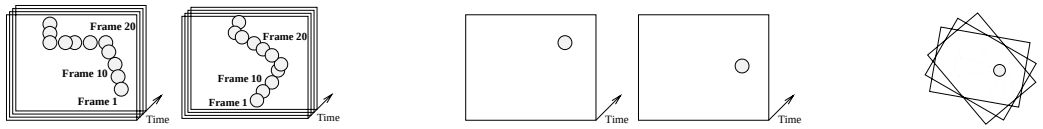
### 3.1 Sequence-to-Sequence vs. Image-to-Image Alignment

This section shows that sequence-to-sequence is a generalization of image-to-image alignment. We first show that when there are no temporal changes in the scene, sequence-to-sequence alignment reduces to image-to-image alignment, with an improved signal-to-noise ratio. In particular it is shown that in such cases, the presented algorithm in Section 2 reduces to the image alignment algorithm of [1].

When there are no temporal changes in the scene, all temporal derivatives within the sequence are zero: $S_t \equiv 0$. Therefore, for any voxel $(x, y, t)$, the error term of Eq. (3) reduces to:

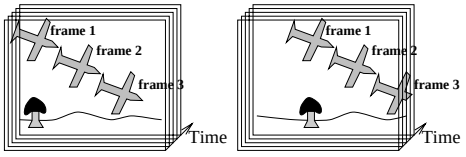
$$\underbrace{e_{seq}(x, y, t; \vec{P}))}_{\text{seq-to-seq}} = S' - S + [u, v]\begin{bmatrix} S_x \\ S_y \end{bmatrix} =$$
$$= I' - I + [u, v]\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \underbrace{e_{img}(x, y; \vec{P})}_{\text{img-to-img}}.$$

where, $I(x, y) = S(x, y, t)$ is the image frame at time $t$.

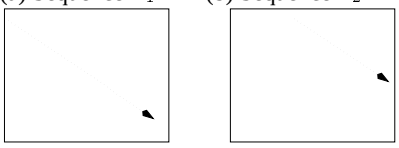



(a) Sequence $S_1$ (b) Sequence $S_2$ (c)Frame from $S_1$ (d)Frame from $S_2$ (e) Possible image alignments

**Figure 2. Spatial ambiguities in image-to-image alignment** *(a) and (b) display two sequences of a moving ball. (c) and (d) show two corresponding frames from the two sequences. There are infinitely many valid image-to-image alignments between the two frames, some of them shown in (e), but only one of then aligns the two trajectories.*




(a) Sequence $S_1$ (b) Sequence $S_2$



(c) Trajectory 1 (d) Trajectory 2

**Figure 3. Spatio-temporal ambiguity in trajectory-to-trajectory alignment** *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 resolves 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 in this case either.*

Therefore, the SSD function of Eq. (2) reduces to:

$$ERR_{seq}(\vec{P}) = \sum_{x,y,t}(e(x,y,t;\vec{P}))^2 =$$
$$= \sum_t \left(\sum_{x,y}(e(x,y,t;\vec{P}))^2\right) = \sum_t ERR_{img}(\vec{P}).$$

namely, the image-to-image alignment objective function, averaged over all frames.

We next show that when the scene *does* contain temporal variations, sequence-to-sequence uses more information for spatial alignment than image-to-image alignment has access to. In particular, there are ambiguous scenarios for image-to-image alignment, which sequence-to-sequence alignment can uniquely resolve. Fig. 2 illustrates a case which is ambiguous for image-to-image alignment. Consider a uniform background scene with a moving ball (Fig. 2.a and Fig. 2.b). At any given frame (e.g., Fig. 2.c and Fig. 2.d) all the spatial gradients are concentrated in a very small image region (the moving ball). In these cases, image-to-image alignment cannot uniquely determine the correct spatial transformation (see Fig. 2.e). Sequence-to-sequence alignment, on the other hand, does not suffer from spatial ambiguities in this case, as the spatial transformation must simultaneously bring into alignment all corresponding frames across the two sequences, i.e., the two trajectories (depicted in Fig. 2.a and Fig. 2.b) must be in alignment.

### 3.2 Sequence-to-Sequence vs. Trajectory-to-Trajectory Alignment

While "trajectory-to-trajectory" alignment can also handle the alignment problem in Fig. 2, there are often cases where analysis of trajectories of temporal information *alone* does *not* suffice to uniquely determine the spatio-temporal transformation between the two sequences. Such is the case in Fig. 3. When only the moving object information is considered (i.e., the trajectory of the airplane), then for any temporal shift, there exists a consistent spatial transformation between the two sequences, which will bring the two trajectories in Figs. 3.c and 3.d into alignment. Namely, in this scenario, trajectory-to-trajectory alignment will find infinitely many valid spatio-temporal transformations. Stein [6] 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. ([3] also reported a similar problem in their formulation).

While trajectory-to-trajectory alignment will find infinitely many valid spatio-temporal transformations for the scenario in Fig. 3, only one of those spatio-temporal transformations will also be consistent with the *static background* (i.e., the tree and the horizon). Sequence-to-sequence alignment will therefore *uniquely* resolve the ambiguity in this case, as it forces both spatial and temporal information to be brought *simultaneously* into alignment across the two sequences.

The direct method for sequence-to-sequence alignment presented in Section 2 is only one possible algorithm for solving this problem. The concept of sequence-to-sequence alignment, however, is more general, and is not limited

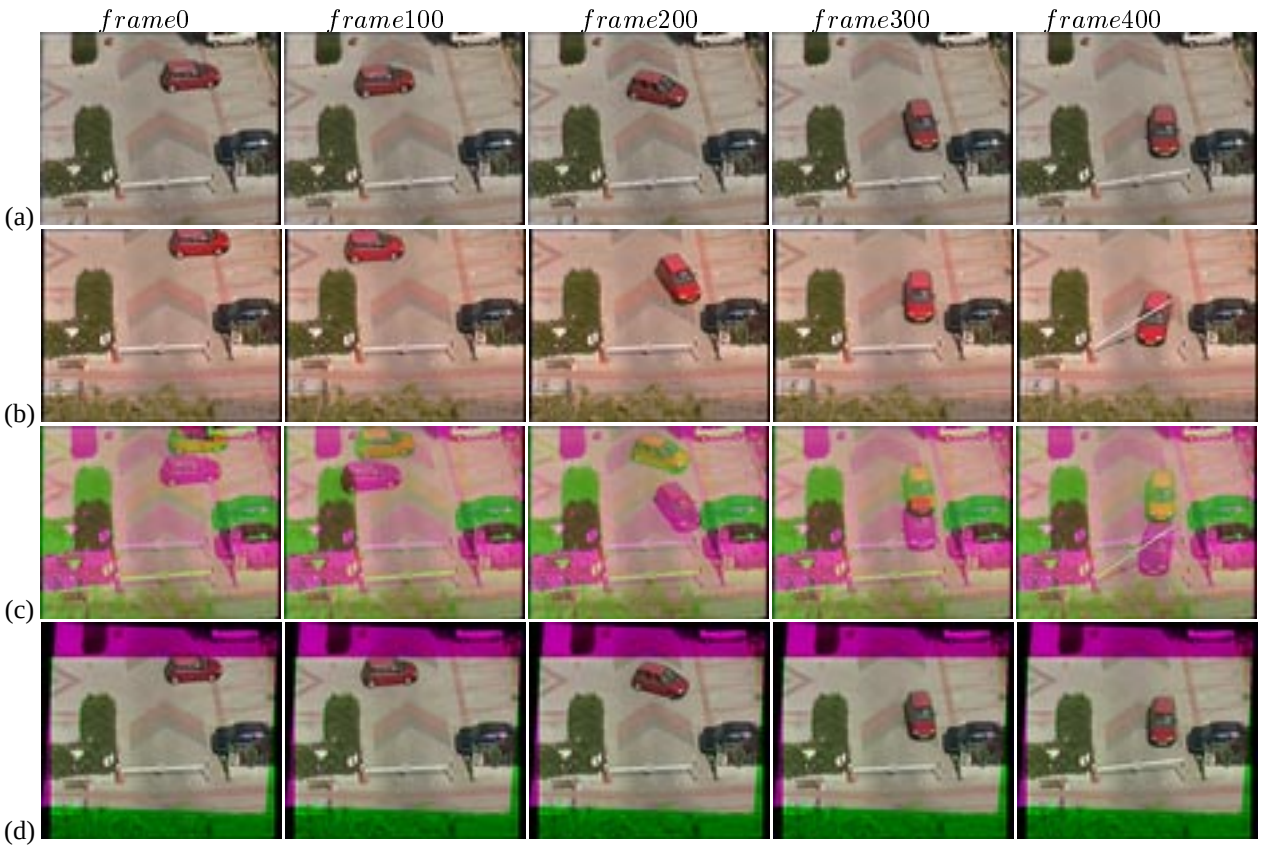



**Figure 4. Scene with moving objects.** *Rows (a) and (b) display five representative frames (0,100,200,300,400) from the reference and second sequences, respectively. The spatial misalignment is easily observed near image boundaries, where different static objects are visible in each sequence. The temporal misalignment is observed 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**

to that particular algorithm. One could, for example, extend the feature-based trajectory-to-trajectory alignment algorithm of [6] into a *feature-based* sequence-to-sequence alignment algorithm, by adding static feature correspondences to the dynamic features.

While feature-based methods can theoretically account for larger spatio-temporal misalignments, it is important to note that the direct method suggested in Section 2 obtains spatio-temporal alignment between the two sequences *without* the need to explicitly separate and distinguish between the two types of information – the spatial and the temporal. Moreover, it does *not* require any explicit detection and tracking of moving objects, nor does it need to detect features and explicitly establish their correspondences across sequences. Finally, because temporal variations need not be explicitly modeled in the direct method, it can exploit other temporal variations in the scene, such as changes in illumination. Such temporal variations are not captured by trajectories of moving objects.

### 3.3 Illumination Changes as a Cue for Alignment

Temporal derivatives are not necessarily a result of independent object motion, but can also result from other changes in the scene which occur over time, such as changes in illumination. Dimming or brightening of the light source are often sufficient to determine the temporal alignment. Furthermore, even homogeneous image regions contribute temporal constraints in this case. This is true although their spatial derivatives are zero, since global changes in illumination produce prominent temporal derivatives.

For example, in the case of the algorithm presented in

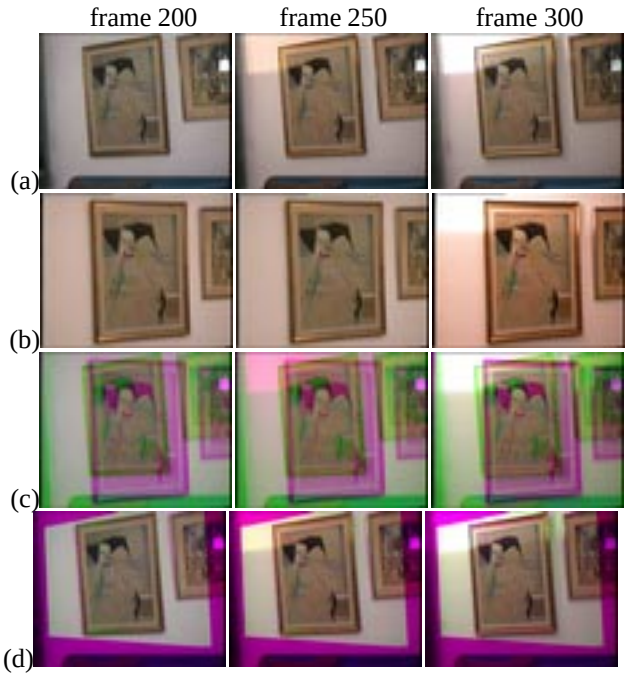



**Figure 5. Scene with varying illumination.** *Rows (a) and (b) display three representative frames (200,250,300) from the reference and second sequences, respectively. The temporal misalignment can be observed in the upper left corner of frame 250, by small differences in illumination. (c) displays 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. 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 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**

Section 2, for a voxel in a uniform region ($S_x = S_y = 0$) undergoing illumination variation ($S_t \neq 0$), Eq. (3) provides the following constraint on the *temporal* alignment parameters: $e(x, y, t; \vec{P}) = (S'(x, y, t) - S(x, y, t)) + w(x, y, t; \vec{P}) S_t(x, y, t)$. Note that, in general, changes in illumination need not be global. For example, an outdoor scene on a partly cloudy day, or an indoor scene with spotlights, can be exposed to local changes in illumination. Such local changes provide additional constraints on the *spatial* alignment parameters. An example of applying our algorithm to sequences with only changes in illumination is shown in Fig. 5.

# 4 Experiments

In our experiments, two different interlaced CCD cameras (mounted on tripods) were used for sequence acquisition. Typical sequence length is several hundreds of frames. Fig. 4 shows a scene with a car driving in a parking lot. When the car reaches the exit, the gate is raised. The two input sequences Figs. 4.a and 4.b were taken from a distance (from two different windows of a tall building). 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 (the sequences were out of synchronization; note the different timing of the gate being lifted in the two sequences), as well as in space (note the misalignment in static scene parts, such as in the other parked cars or at the bushes). Fig. 4.d shows the superposition of frames *after* applying spatio-temporal alignment. The second sequence was spatio-temporally warped towards the reference sequence according to the computed parameters. The recovered temporal shift was 46.5 frames, and was verified against the ground truth, obtained by auxiliary equipment. The recovered spatial affine transformation indicated a translation on the order of a $1/5$ of the image size, a small rotation, a small scaling, and a small skew (due to different aspect ratios of the two cameras). Note the good quality of alignment despite the overall difference in chroma and brightness between the two input sequences.

Fig. 5 illustrates that temporal alignment is not limited to motion information alone. A light source was brightened and then dimmed down, resulting in observable illumination variations in the scene. The cameras were imaging a picture on a wall from significantly different viewing angles, inducing a significant perspective distortion. Fig. 5.a and 5.b show a few representative frames from two sequences of several hundred frames each. The effects of illumination are particularly evident in the upper left corner of the image. Fig. 5.c shows a superposition of the representative frames from both sequences *before* spatio-temporal alignment. Fig. 5.d shows superposition of corresponding frames *after* spatio-temporal alignment. The recovered temporal offset (21.3 frames) was verified against the ground truth. 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 temporal alignment (frame 250.d). The reader is encouraged to view full color sequences at www.wisdom.weizmann.ac.il/Seq2Seq

# 5 Conclusion and Future Work

In this paper we have introduced a new approach to sequence-to-sequence alignment, which simultaneously

uses all available spatial and temporal information within the video sequences. We showed that our approach combines the benefits of image-to-image alignment with the benefits of trajectory-to-trajectory alignment, and is a generalization of both approaches. Furthermore, it resolves many of the inherent ambiguities associated with each of these two classes of methods.

The current discussion and implementation were restricted to stationary cameras, and hence used only two types of information cues for alignment - the *scene dynamics* and the *scene appearance*. We are currently extending our approach to handle moving cameras. This adds a third type of information cue for alignment, which is inherent to the scene and is common to the two sequences - the *scene geometry*.

While the approach is general, we have also presented a specific algorithm for sequence-to-sequence alignment, which recovers the spatio-temporal alignment parameters directly from spatial and temporal brightness variations within the sequence. However, the paradigm of sequence-to-sequence alignment extends beyond this particular algorithm and beyond direct methods. It can equally employ feature-based matching across sequences, or other type of match measures (e.g., mutual information).

## References


[1] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, 1992.

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

[3] M. A. Giese and T. Poggio. Synthesis and recognition of biological motionpatterns on linear superposition prototypical motion sequences. In *International Conference on Computer Vision*, pages 73–80, 1998.

[4] 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.

[5] I. Reid and A. Zisserman. Goal-directed video metrology. In *European Conference on Computer Vision*, pages 647–658, 1996.

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

[7] E. Grimson P. Viola O.Faugeras T. Lozano-Perez T. Poggio S. Teller. A forest of sensores. In *International Conference on Computer Vision*, pages 45–51, 1997.


## Appendix A: 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 induces 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

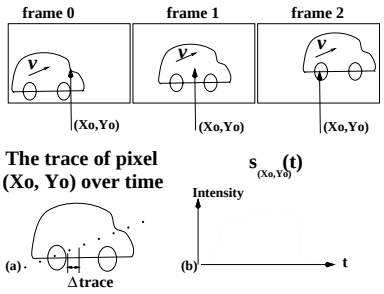



**Figure 6. Induced temporal frequencies.** *Three frames 0,1,2 of a car moving up right with velocity $v$ are presented above. A fixed pixel $(x_0, y_0)$ is marked on each frame. (a) displays the trace of the pixel. (b) displays the gray level values along this trace.*

along the trace (See Fig. 6). Denote by $\Delta trace$ the spatial step size along the trace. For an object moving at velocity v: $\Delta trace = v\Delta t$, where $\Delta t$ is the time difference between two successive frames. To avoid temporal aliasing, $\Delta trace$ must satisfy the Shannon-Whittaker sampling theorem: $\Delta trace <= \frac{1}{2\omega}$, where $\omega$ 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 $\omega$ 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 (Section 2.3) 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 its own size.

## Acknowledgment


The authors would like to thank P. Anandan and L. Zelnik-Manor for their helpful comments.

# 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.

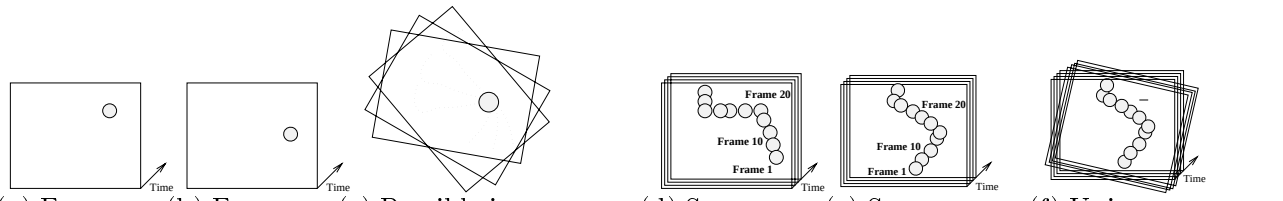
2. The ability to align and integrate information across multiple sequences both in space and in time gives rise to new video applications that are not possible when only image-to-image alignment is used.

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.

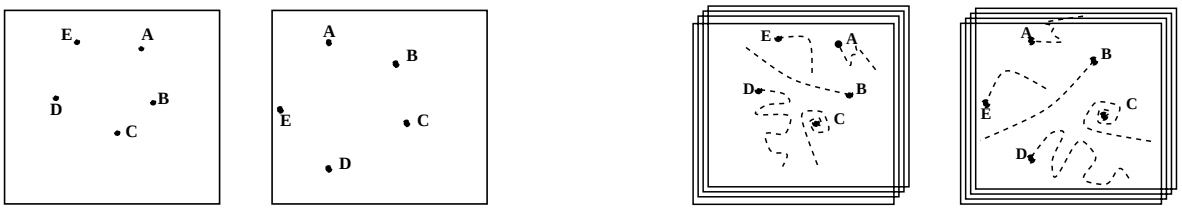
The rest of the paper is organized as follows: In Section 2 we formulate the problem of sequence-to-sequence alignment. In Section 3 we present two sequence-to-sequence alignment algorithms (the feature-based and the direct-based). Section 4 discusses the properties of sequence-to-sequence alignment, 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 $3 \times 3$ 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 $i$th 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) :

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

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 $\Delta t$ and approximate $H$ using standard methods (e.g., the DLT algorithm described in [15]).
(ii) Fix $H$ and refine $\Delta t$ 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:

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

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

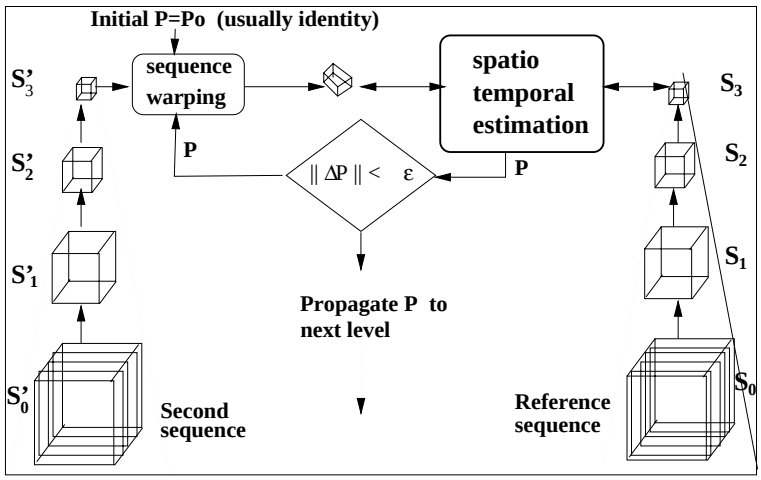



Figure 3: **Direct sequence-to-sequence alignment.** *A spatio-temporal pyramid is constructed for each input sequence: one for the reference sequence (on the right side), and one for the second sequence (on the left side). The spatio-temporal alignment estimator is applied iteratively at each level. 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 $\vec{P}$:

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

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 $256 \times 256$ pixels, then the next resolution level contains a sequence of 32 frames of size $128 \times 128$, 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.

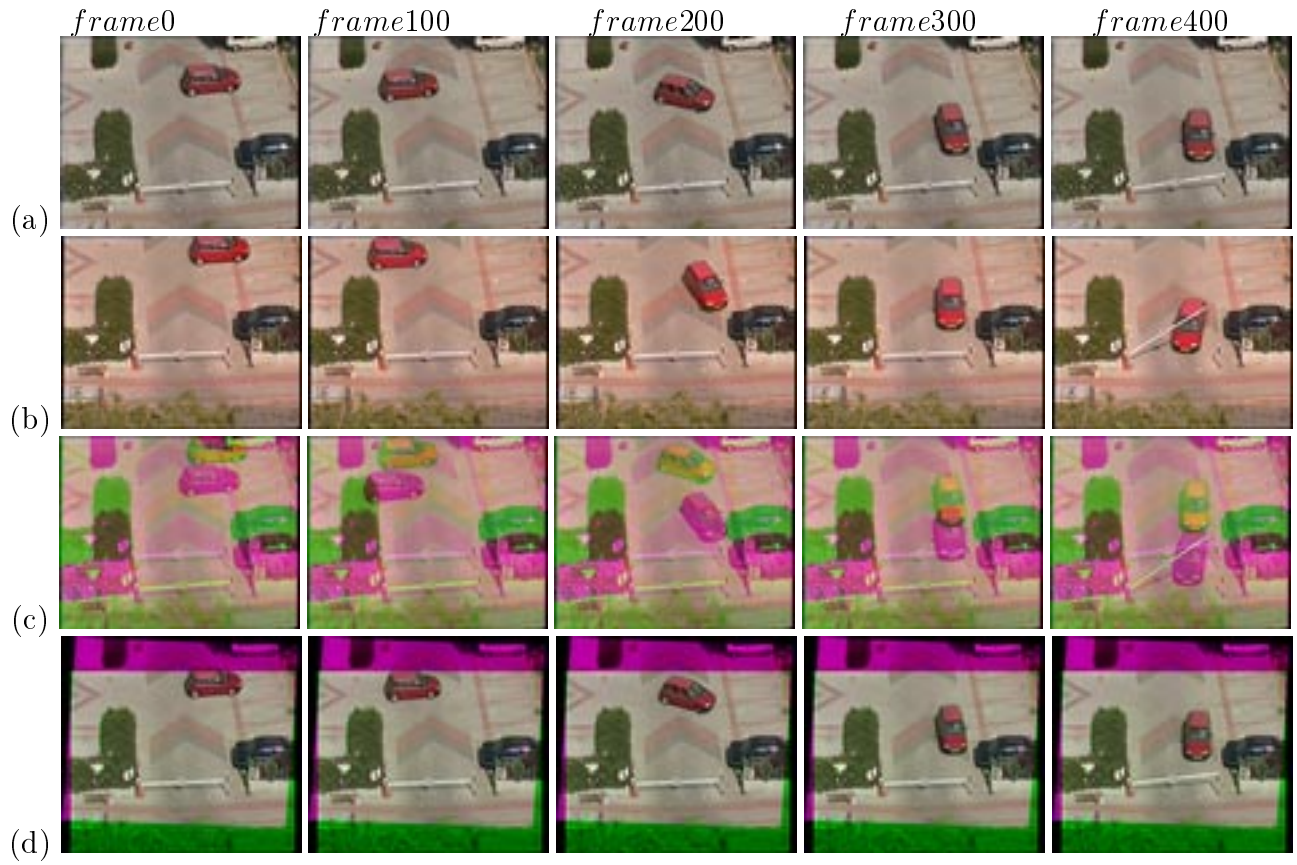



Figure 4: **Scene with moving objects.** *Rows (a) and (b) display five representative frames (0,100,200,300,400) from the reference and second sequences, respectively. The spatial misalignment is easily noticeable near image boundaries, where different static objects are visible in each sequence (e.g., 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.**

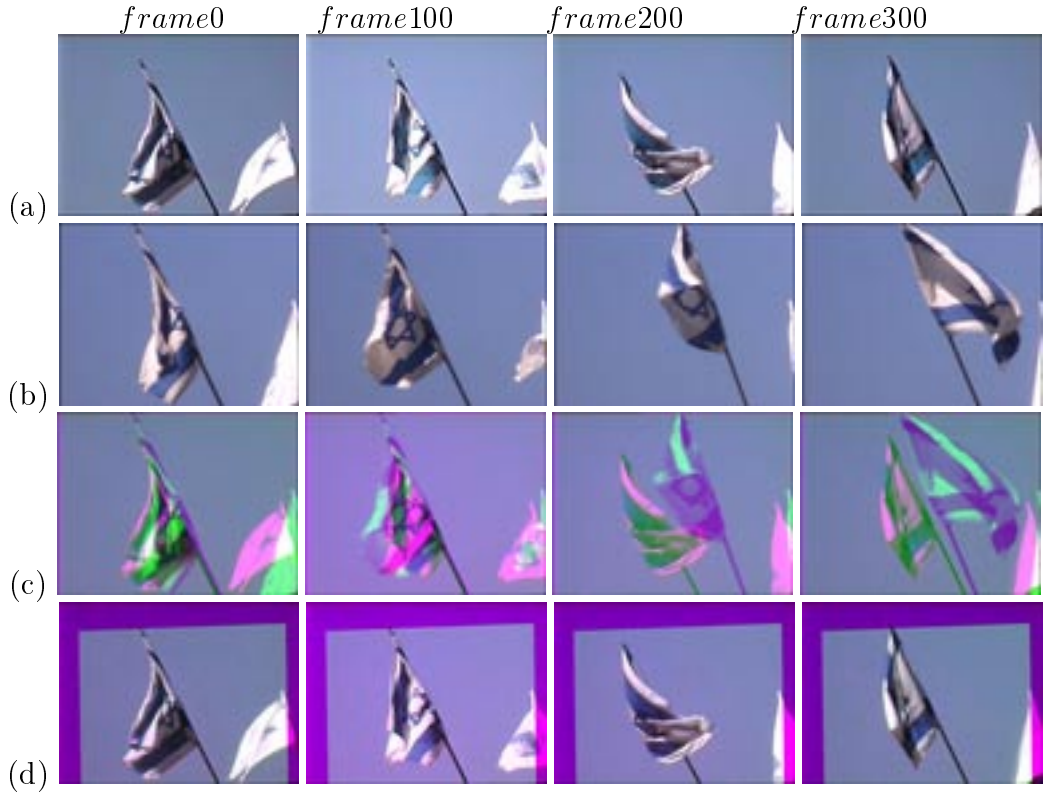



Figure 5: **Scene with non rigid motion.** *Rows (a) and (b) display four representative frames (0,100,200,300) from the reference and second sequences, respectively. Row (c) displays superposition of the representative frames* before *spatio-temporal alignment. The spatial misalignment between the sequences 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.

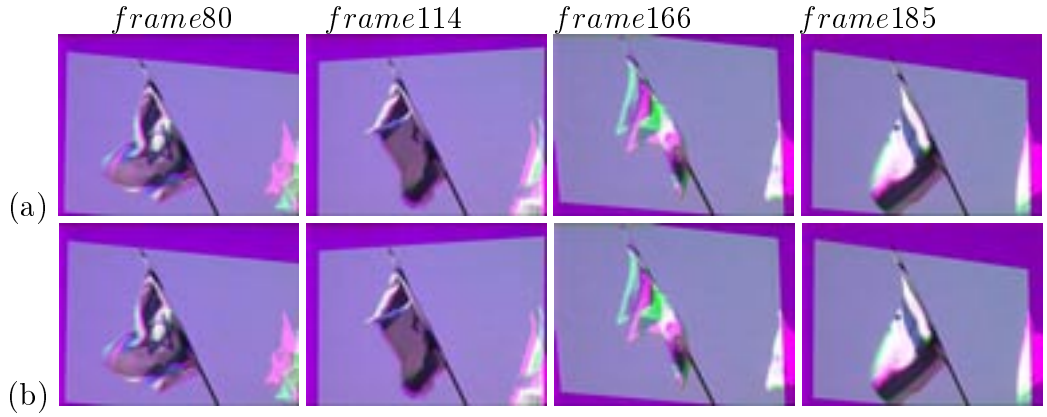



Figure 6: **Image-to-Image alignment vs. Sequence-to-Sequence alignment** *(a) Results of applying image-to-image alignment to temporally corresponding frames. Spatial alignment is inaccurate due to insufficient spatial information in any of these individual frames. (b) Accurate alignment of the same 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).

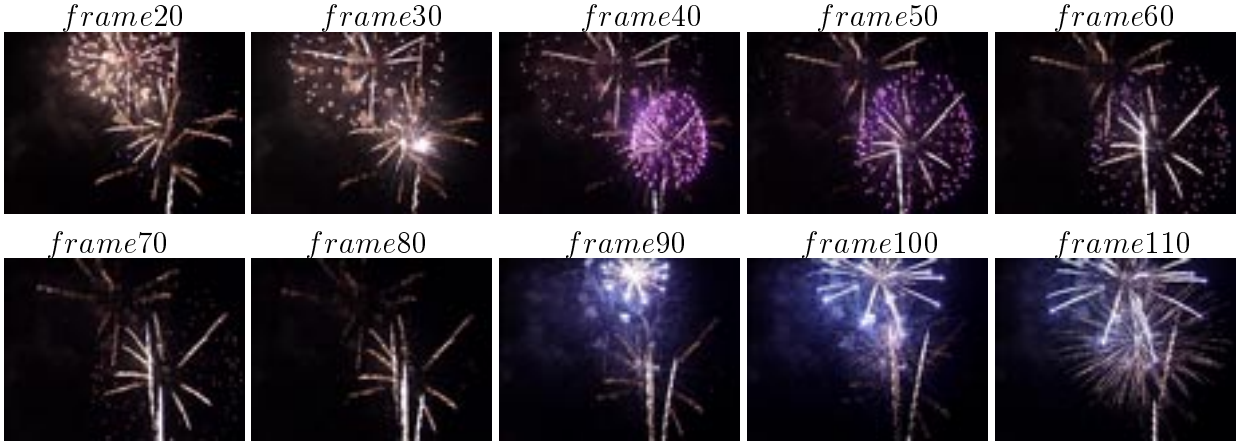
The example in Fig. 4 is rich in spatial texture. Image-to-image alignment therefore also provides high quality *spatial* alignment in this case (when applied to corresponding frames in time across the two sequences). However, this is not the case for the next example. Fig. 5 shows two sequences (5.a 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

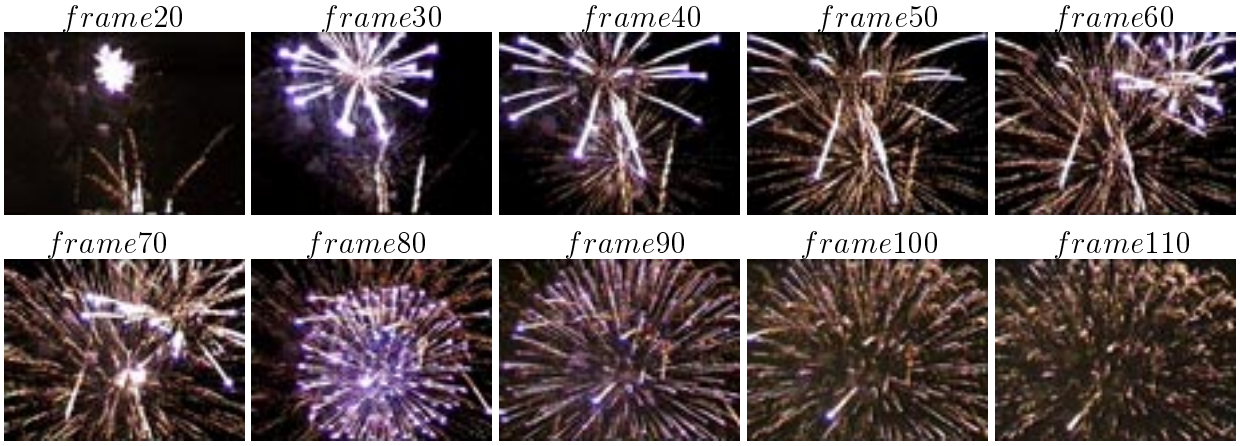
## 4.1 Benefits of Sequence Alignment over Image Alignment

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:

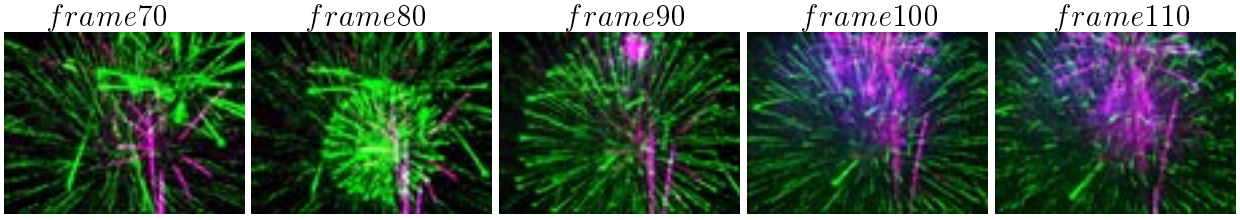
**(a) Reference Sequence:**




**(b) Second Sequence:**




**(c) Before Alignment:**

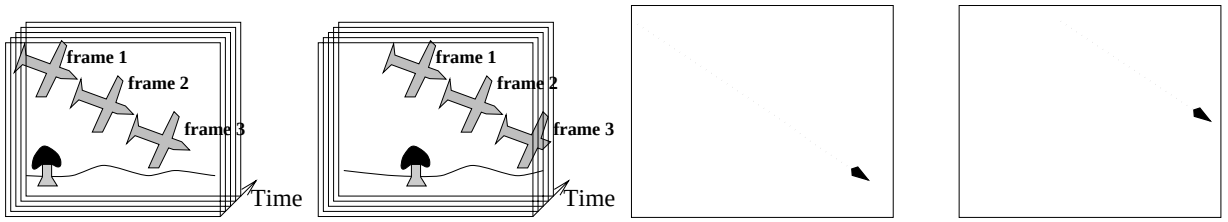



**(d) After Alignment:**


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-

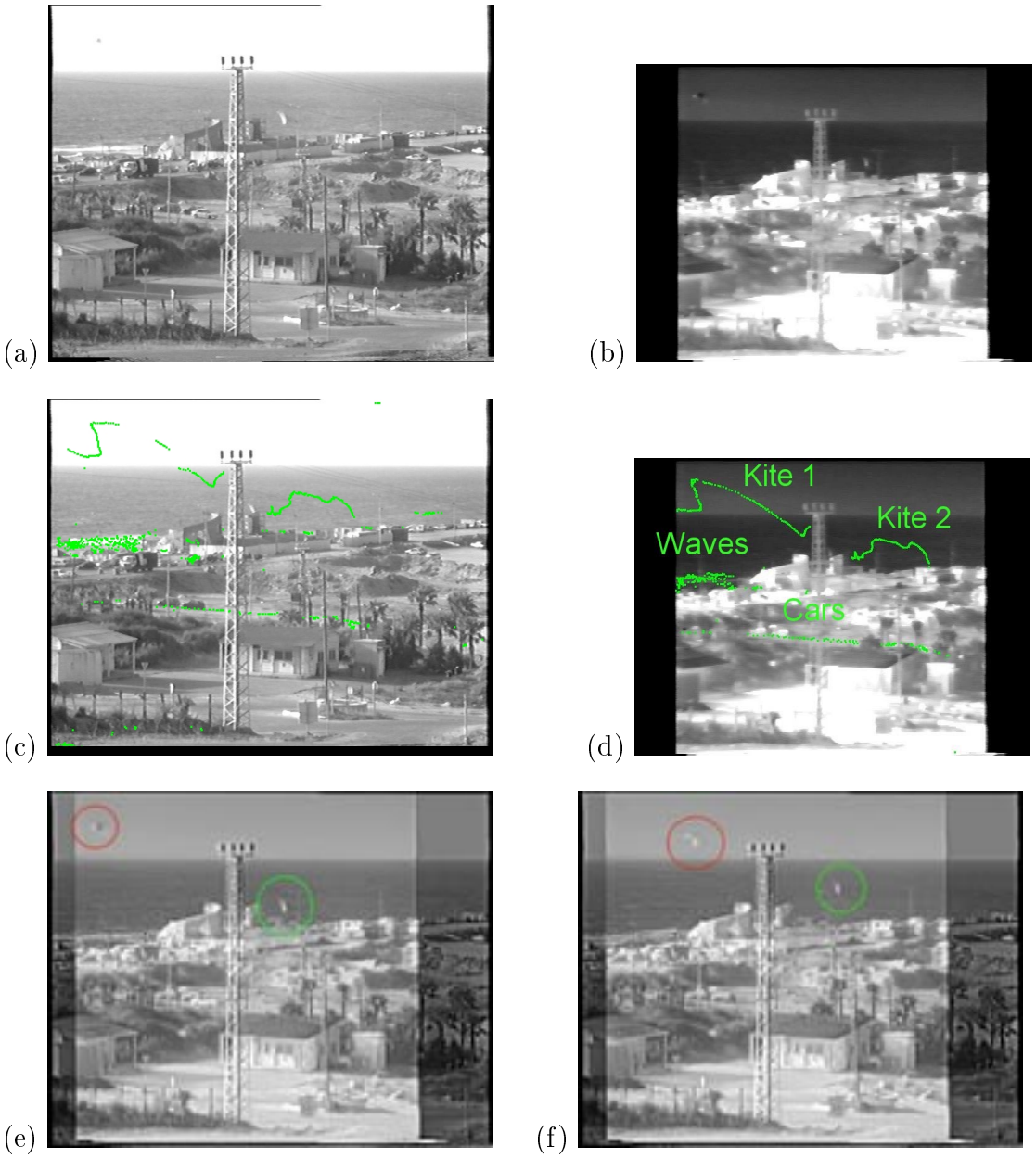



Figure 10: **Multi-Sensor Alignment.** *(a) and (b) display representative frames from a PAL visible light sequence and an NTSC Infra-Red sequence, respectively. The scene contains several moving objects: 2 kites, 2 moving cars, and sea waves. The trajectories induced by tracking the moving objects are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

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

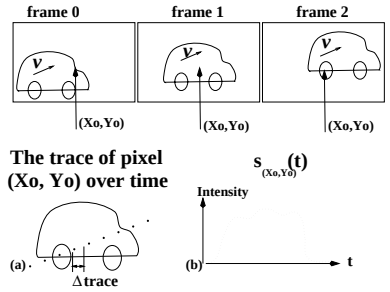



Figure 11: **Induced temporal frequencies.** *Three frames 0,1,2 of a car moving up right with velocity v are presented above. A fixed pixel $(x_0, y_0)$ is marked on each frame. (a) displays the trace of the pixel. (b) displays the gray level values along this trace.*

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 $\nabla S$ 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 $\delta t$ 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 $\omega$ 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 $\omega$ 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 $\Delta t$, then the time shift between $Even_1$ and $Even_2$ should be also $\Delta t$, 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:

$$\underbrace{ERR(\vec{P})}_{\text{seq-to-seq}} = \sum_{x,y,t} S - S' + \begin{bmatrix} S'_x & S'_y & 0 \end{bmatrix} \begin{bmatrix} J_{spatial} & 0 \\ 0 & J_{temporal} \end{bmatrix} \begin{bmatrix} P_{spatial} \\ P_{temporal} \end{bmatrix} =$$
$$= \sum_t \left( \sum_{x,y} I' - I + [I_x \; I_y] \, J_{spatial} \vec{P}_{spatial} \right) = \sum_t \underbrace{err(\vec{P}_{spatial})}_{\text{img-to-img}}$$

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.**

# Alignment of Non-Overlapping Sequences

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

*This paper shows how two image sequences that have no spatial overlap between their fields of view can be aligned both in time and in space. Such alignment is possible when the two cameras are attached closely together and are moved jointly in space. The common motion induces "similar" changes over time within the two sequences. This correlated temporal behavior, is used to recover the spatial and temporal transformations between the two sequences. The requirement of "coherent appearance" in standard image alignment techniques, is therefore replaced by "coherent temporal behavior", which is often easier to satisfy.*

*This approach to alignment can be used not only for aligning non-overlapping sequences, but also for handling other cases that are inherently difficult for standard image alignment techniques. We demonstrate applications of this approach to three real-world problems: (i) alignment of non-overlapping sequences for generating wide-screen movies, (ii) alignment of images (sequences) obtained at significantly different zooms, for surveillance applications, and, (iii) multi-sensor image alignment for multi-sensor fusion.*

## 1 Introduction

The problem of image alignment (or registration) has been extensively researched, and successful approaches have been developed for solving this problem. Some of these approaches are based on matching extracted local image features. Other approaches are based on directly matching image intensities. A review of some of these methods can be found in [16] and [10]. However, all these approaches share one basic assumption: that there is sufficient overlap between the two images to allow extraction of common image properties, namely, that there is sufficient "similarity" between the two images ("Similarity" of images is used here in the broadest sense. It could range from gray-level similarity, to feature similarity, to similarity of frequencies, and all the way to statistical similarity such as mutual information [17]).

In this paper the following question is addressed: *Can two images be aligned when there is very little similarity between them, or even more extremely, when there is no spatial overlap at all between the two images?* When dealing with individual images, the answer tends to be "No". However, this is not the case when dealing with image sequences. An image sequence contains much more information than any individual frame does. In particular, temporal changes (such as dynamic changes in the scene, or the induced image motion) are encoded *between* video frames, but do not appear in any individual frame. Such information can form a powerful cue for alignment of two (or more) sequences. Caspi and Irani [4] and Stein [15] have illustrated the applicability of such an approach for aligning two sequences. However, they assumed that the same temporal changes in the scene (e.g., moving objects) are visible to both video cameras, leading to the requirement that there must be significant overlap in the FOV's (fields-of-view) of the two cameras.

In this paper we show that when two cameras are attached closely to each other (so that their centers of projections are very close), and move jointly in space, then the induced frame-to-frame transformations *within* each sequence have correlated behavior *across* the two sequences. This is true even when the sequences have no spatial overlap. This correlated temporal behavior is used to recover both the spatial and temporal transformations between the two sequences.

Unlike carefully calibrated stereo-rigs [14], our approach does not require any prior internal or external camera calibration, nor any sophisticated hardware. Our approach bears resemblance to the approaches suggested by [5, 9, 18] for auto-calibration of stereo-rigs. But unlike these methods, we do not require that the two cameras observe and match the same scene features, nor that their FOV's will overlap.

The need for "coherent appearance", which is a fundamental assumption in image alignment methods, is replaced here with the requirement of "coherent temporal behavior". This requirement is often easier to satisfy (e.g., by moving the two cameras jointly in space). Our approach is therefore useful not only in the case of non-overlapping sequences, but also in other cases which are inherently difficult for standard image alignment techniques.

This gives rise to a variety of real-world applications, including: (i) Multi-sensor alignment for image fusion. This requires accurate alignment of images (sequences) obtained

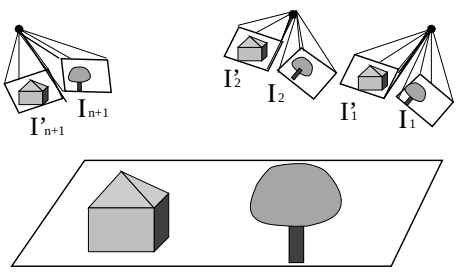



Figure 1: *Two video cameras are attached to each other, so that they have the same center of projection, but non-overlapping fields-of-view. The two cameras are moved jointly in space, producing two separate video sequences* $I_1, ..., I_{n+1}$ *and* $I'_1, ..., I'_{n+1}$.

by sensors of different sensing modalities (such as Infra-Red and visible light). Such images differ significantly in their appearance due to different sensor properties [17]. (ii) Alignment of images (sequences) obtained at different zooms. The problem here is that different image features are prominent at different image resolutions [6]. Alignment of a wide-FOV sequence with a narrow-FOV sequence is useful for detecting small zoomed-in objects in (or outside) a zoomed-out view of the scene. This can be useful in surveillance applications. (iii) Generation of wide-screen movies from multiple non-overlapping narrow FOV movies (such as in IMAX movies).

Our approach can handle such cases. Results are demonstrated in the paper on complex real-world sequences, as well as on manipulated sequences with ground truth.

## 2 Problem Formulation

We examine the case when two video cameras having (approximately) the same center of projection, but different 3D orientation, move jointly in space (see Fig. 1). The fields of view of the two cameras do not necessarily overlap. The internal parameters of the two cameras are different and unknown, but fixed along the sequences. The external parameters relating the two cameras (i.e., the relative 3D orientation) are also unknown but fixed. Let $S = I_1, ...I_{n+1}$ and $S' = I'_1, ..., I'_{m+1}$ be the two sequences of images recorded by the two cameras[1]. When temporal synchronization (e.g., time stamps) is not available, then $I_i$ and $I'_i$ may not be corresponding frames in time. Our goal is to recover the transformation that aligns the two sequences both in time and in space. Note the term "alignment" here has a broader meaning than the usual one, as the sequences may not overlap in space, and may not be synchronized in time. Here we refer to alignment as displaying one sequence in the spatial coordinate system of the other sequence, and at the correct time shift, as if obtained by the other camera.

When the two cameras have the same center of projection, then a simple fixed homography $H$ (a 2D projective transformation) describes the *spatial* transformation between temporally corresponding pairs of frames across the two sequences. This is shown next.

Let $C$ and $C'$ denote the two 3x3 matrices capturing the internal parameters of the two cameras. Let $p = [x, y, 1]^t$ and $p' = [x', y', 1]^t$ be 2D image points in the two cameras which correspond to the same 3D point and the same time ($[]^t$ denotes the transpose). In the general case [8]:

$$p' \cong C'[RC^{-1}p + \gamma\vec{t}] \quad (1)$$

where $R$ is a 3x3 matrix capturing the relative orientation of the two cameras, $\gamma$ is a scale factor, $\vec{t}$ is the 3D translation between the two cameras, and $\cong$ denotes equality up to scale factor. When the two cameras have the same center of projection, then the translation between the two cameras is zero: $\vec{t} = 0$. (In practice, $\vec{t}$ will be negligible, but not 0.) The relation of Eq. (1) therefore reduces to: $p' \cong Hp$ where $H = C'RC^{-1}$ is a homography. As the internal parameters $C$ and $C'$ and the relative orientation $R$ are fixed, the homography $H$ is also fixed along the sequence. This homography (which is unknown) relates every pair of temporally corresponding frames across the two sequences.

If there were enough common features (e.g., $p$ and $p'$) between temporally corresponding frames (e.g., $I_i$ and $I'_i$), then it would be easy to recover the inter-camera homography H, as each such pair of corresponding image points provides two linear constrains on H: $p' \cong Hp$. This, in fact, is how most image alignment techniques work [8]. However, this is not the case here. The two sequence do not share common features, because there is no spatial overlap between the two sequences. Instead, the homography H is recovered from the induced frame-to-frame transformations *within* each sequence.

Let $T_1, ...T_n$ and $T'_1, ...T'_m$ be the sequences of frame-to-frame transformations within the video sequences $S$ and $S'$, respectively. $T_i$ is the transformation relating frame $I_i$ to $I_{i+1}$. These transformations can be either 2D parametric transformations (e.g., homographies or affine transformations) or 3D transformations/relations (e.g., fundamental matrices). We next show how we can recover the spatial transformation H and the temporal shift $\Delta t$ between the two video sequences directly from the two sequences of transformations $T_1, ...T_n$ and $T'_1, ...T'_m$. The problem formulated above is illustrated in Fig 2.

[1]The subscript i is used represents the frame time index, and the superscript prime is used to distinguish between the two sequences $S$ and $S'$.

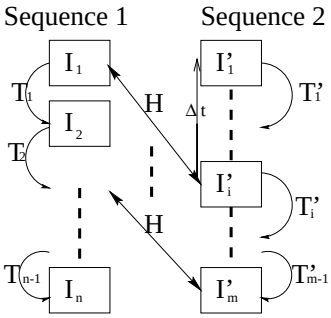



Figure 2: **Problem formulation.** *The two sequences are spatially related by a fixed but unknown inter-camera homography H, and temporally related by a fixed and unknown time shift $\Delta t$. Given the frame-to-frame transformations $T_1, ..., T_n$ and $T'_1, ..., T'_m$, we want to recover $H$ and $\Delta t$.*

# 3 Recovering Spatial Alignment Between Sequences

Let us first assume that the temporal synchronization is known. Such information is often available (e.g., from time stamps encoded in each of the two sequences). Section 4 shows how we can recover the temporal shift between the two sequences when that information is not available. Therefore, without loss of generality, it is assumed that $I_i$ and $I'_i$ are corresponding frames in time in sequences $S$ and $S'$, respectively. Two cases are examined: (i) The case when the scene is planar or distant from the cameras. We refer to these scenes as "2D scenes". In this case the frame-to-frame transformations $T_i$ can be modeled by homographies (Sec. 3.1). (ii) The case of a non-planar scene. We refer to these scenes as "3D scenes". In this case the frame-to-frame relation can be modeled by a fundamental matrix (Sec. 3.2).

## 3.1 Planar or Distant (2D) Scenes

When the scene is planar or distant from the cameras, or when the joint 3D translation of the two cameras is negligible relative to the distance of the scene, then the induced image motions within each sequence (i.e., $T_1, ...T_n$ and $T'_1, ...T'_n$) can be described by 2D parametric transformations [8]. $T_i$ denotes the homography between frame $I_i$ and $I_{i+1}$, represented by $3 \times 3$ non-singular matrices. We next show that temporally corresponding transformations $T_i$ and $T'_i$ are also related by the fixed inter-camera homography $H$.

Let $P$ be a 3D point in the planar (or the remote) scene. Denote by $p_i$ and $p'_i$ its image coordinates in frames $I_i$ and $I'_i$, respectively (the point $P$ need not to be visible in the frames, i.e., $P$ need not be within the FOV of the cameras). Let $p_{i+1}$ and $p'_{i+1}$ be its image coordinates in frames $I_{i+1}$ and $I'_{i+1}$, respectively. Then, $p_{i+1} \cong T_i p_i$ and $p'_{i+1} \cong T'_i p'_i$. Because the coordinates of the video sequences S and S' are related by a fixed homography $H = C'RC^{-1}$ (see Sec. 2), then: $p' \cong Hp$ and $p'_{i+1} \cong Hp_{i+1}$. Therefore:

$$HT_i p_i \cong Hp_{i+1} \cong p'_{i+1} \cong T'_i p'_i \cong T'_i H p_i \qquad (2)$$

Each $p_i$ could theoretically have a different scalar associated with the equality in Eq. (2). However, it is easy to show that because the relation in Eq. (2) holds for *all* points $p_i$, therefore all these scalars are equal, and hence:

$$HT_i \cong T'_i H. \qquad (3)$$

Because $H$ is invertible, we may write $T'_i \cong HT_iH^{-1}$, or

$$T'_i = s_i HT_iH^{-1} \qquad (4)$$

where $s_i$ is a (frame-dependent) scale factor. Eq. (4) is true for all frames (i.e., for any pair of corresponding transformations $T_i$ and $T'_i$, $i = 1..n$). Eq. (4) shows that there is a **similarity relation**[2] between the two matrices $T_i$ and $T'_i$ (up to a scale factor). A similar observation was made by [18] for the case of auto-calibration of a stereo-rig.

Denote by $eig(A) = [\lambda_1, \lambda_2, \lambda_3]^t$ a $3 \times 1$ vector containing the eigenvalues of a $3 \times 3$ matrix $A$ (in decreasing order). Then it is known ([7] pp. 898.) that: (i) If $A$ and $B$ are similar matrices, then they have the same eigenvalues: $eig(A) = eig(B)$, and, (ii) The eigenvalues of a scaled matrix are scaled: $eig(sA) = s(eig(A))$. Using these two facts and Eq. (4) yields:

$$eig(T'_i) = s_i \; eig(T_i) \qquad (5)$$

where $s_i$ is the scale factor defined by Eq. (4). Eq. (5) implies that $eig(T_i)$ and $eig(T'_i)$ are "parallel". This gives rise to a measure of similarity between two matrices $T_i$ and $T'_i$:

$$sim(T_i, T'_i) = \frac{eig(T_i)^t \; eig(T'_i)}{||eig(T_i)|| \; ||eig(T'_i)||}, \qquad (6)$$

where $||\cdot||$ is the vector norm. For real valued eigenvalues, Eq. (6) provides the cosine of the angle between the two vectors $eig(T_i)$ and $eig(T'_i)$. This property will be used later for obtaining the temporal synchronization (Section 4). This measure is also used for outlier rejection of bad frame-to-frame transformation pairs, $T_i$ and $T'_i$. The remainder of this section explains how the fixed inter-camera homography H is recovered from the list of frame-to–frame transformations $T_1, ..T_n$ and $T'_1, .., T'_n$, and discusses uniqueness of the solution.

For each pair of temporally corresponding transformations $T_i$ and $T'_i$ in sequences $S$ and $S'$, we first compute their eigenvalues $eig(T_i)$ and $eig(T'_i)$. The scale factor $s_i$ which relates them is then estimated from Eq. (5) using least squares minimization. (three equations one unknown). Once $s_i$ is estimated, Eq. (4) (or Eq. (3)) can be rewritten as:

$$s_i HT_i - T'_i H = 0 \qquad (7)$$

[2] A matrix $A$ is said to be "similar" to a matrix $B$ if there exists an invertible matrix $M$ such that $A = MBM^{-1}$. See [7].

Eq. (7) is linear in the unknown components of $H$. Rearranging the components of $H$ in a $9 \times 1$ column vector $\vec{h} = [H_{11}H_{12}H_{13}H_{21}H_{22}H_{23}H_{31}H_{32}H_{33}]^t$, Eq. (7) can be rewritten as a set of linear equations in $\vec{h}$:

$$M_i\vec{h} = \vec{0} \tag{8}$$

where $M_i$ is a $9 \times 9$ matrix defined by $T_i$, $T_i'$ and $s_i$:

$$M_i = \left[\begin{array}{c|c|c} s_i{T_i}^t - T'_{i_{11}}I & -T'_{i_{12}}I & -T'_{i_{13}}I \\ \hline -T'_{i_{21}}I & s_iT^t - T'_{i_{22}}I & -T'_{i_{23}}I \\ \hline -T'_{i_{31}}I & -T'_{i_{32}}I & s_iT^t - T'_{i_{33}}I \end{array}\right]_{9\times 9}$$

and $I$ is the $3 \times 3$ identity matrix.

Eq. (8) implies that each pair of corresponding transformations $T_i$ and $T_i'$ contributes 9 linear constrains in the unknown homography $H$ (i.e., $\vec{h}$). It can be shown that if $T_i$ (and hence also $T_i'$) have 3 *different* eigenvalues, then $H$ can be determined by a single such pair of transformations up to three degrees of freedom. Therefore, at least two such pairs of independent transformations are needed to uniquely determine the homography $H$ (up to a scale factor).

The constraints from all the transformations $T_1, .., T_n$ and $T_1', .., T_n'$ can be combined into a single set of linear equations in $\vec{h}$:

$$A\vec{h} = \vec{0} \tag{9}$$

where A is a $9n \times 9$ matrix: $A = \left[\begin{array}{c} M_1 \\ \vdots \\ M_n \end{array}\right]$. Eq. (9) is a homogeneous set of linear equations in $\vec{h}$, that can be solved in a variety of ways [2]. In particular, $\vec{h}$ may be recovered by computing the eigenvector which corresponds to the smallest eigenvalue of the matrix $A^tA$.

## 3.2 3D Scenes

When the scene is neither planer nor distant, the relation between two consecutive frames of an uncalibrated camera is described by the fundamental matrix [8]. In this case the input to our algorithm is two sequences of fundamental matrices between successive frames, denoted by $F_1, ...F_n$ and $F_1', ...F_n'$. Namely, if $p_i \in I_i$ and $p_{i+1} \in I_{i+1}$ are corresponding image points, then: $p_{i+1}^t F_i p_i = 0$. Although the relations *within* each sequence are characterized by fundamental matrices, the inter-camera transformation remains a homography $H$. This is because the two cameras still share the same center of projection (Sec. 2).

Each fundamental matrix $F_i$ can be decomposed into a homography + epipole as follows [8]:

$$F_i = [e_i]_x T_i$$

Figure 3: **Alignment of non-overlapping sequences.** *(a) and (b) are temporally corresponding frames from sequences $S$ and $S'$. The correct time shift was automatically detected. (c) shows one frame in the combined sequence after spatio-temporal alignment. Note the accuracy of the spatial and temporal alignment of the running person.* **For full color sequences see attached tar file.**

where $e_i$ is the epipole relating frames $I_i$ and $I_{i+1}$, the matrix $T_i$ is the induced homography from $I_i$ to $I_{i+1}$ via any plane (real or virtual). $[\cdot]_x$ is the cross product matrix ($[v]_x\vec{w} = \vec{v} \times \vec{w}$).

The homographies, $T_1, ..., T_n$ and $T_1', ..., T_n'$, and the epipoles $e_1, ..., e_n$ and $e_1', ..., e_n'$, impose separate constraints on the inter-camera homography $H$. These constraints can be used separately or jointly to recover $H$.

**(i) Homography-based constraints:** The homographies $T_1, .., T_n$ and $T_1', .., T_n'$ (extracted from the fundamental matrices $F_1, .., F_n$ and $F_1', .., F_n'$, respectively), may correspond to different 3D planes. In order to apply the algorithm of Sec. 3.1 using these homographies, we need impose plane-consistency across the two sequences (to guarantee that temporally corresponding homographies correspond to the same plane in the 3D world). One possible way for imposing plane-consistency across (and within) the two sequences is by using the "Plane+Parallax" approach [13, 11]. However, this approach requires that a real physical planar surface be visible in *all* video frames. Alternatively, the "threading" method of [1] can impose plane-consistency within each sequence, even if no real physical plane is visible in any of the frames. Plane consistency *across* the two sequences can be guaranteed e.g., if [1] is initiated at frames which are known to simultaneously view the same real plane in both sequences. However, the two cameras can see different portions of the plane (allowing for non-overlapping FOVs), and need not see the plane at any of the other frames. This approach is therefore less restrictive than

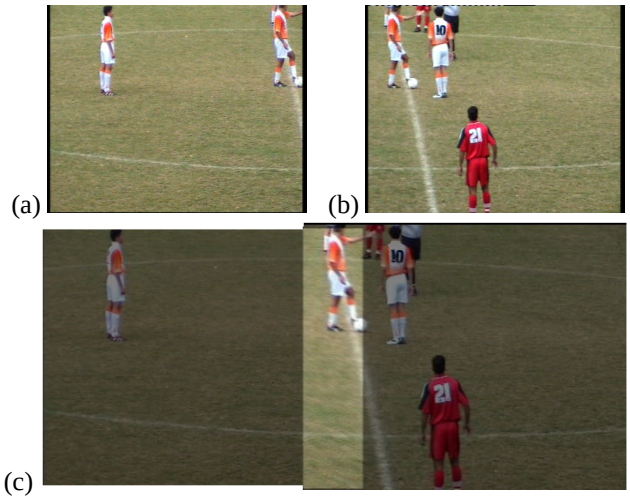



Figure 4: **Alignment of non-overlapping sequences.** *(a) and (b) are temporally corresponding frames from sequences $S$ and $S'$. The correct time shift was automatically detected. (c) shows one frame in the combined sequence. Corresponding video frames were averaged after spatio-temporal alignment. The small overlapping area was* **not** *used in the estimation process, but only for verification (see text). Note the accuracy of the spatial and temporal alignment of the soccer player in the overlapping region.* **For the natural-looking wide-screen sequence see attached tar file.**

the Plane+Parallax approach.

**(ii) Epipole-based constraints:** The fundamental matrices $F_1..F_n$ and $F'_1..F'_n$ also provide a list of epipoles $e_1, ..., e_n$ and $e'_1, ..., e'_n$. These epipoles are uniquely defined (there is no issue of plane consistency here). Since the two cameras have the same center of projection, then for any frame $i$: $e'_i \cong He_i$, or more specifically:

$$(e'_i)_x = \frac{[h_1 h_2 h_3]\, e_i}{[h_7 h_8 h_9]\, e_i} \quad (e'_i)_y = \frac{[h_4 h_5 h_6]\, e_i}{[h_7 h_8 h_9]\, e_i} \tag{10}$$

Multiplying by the dominator and rearranging terms yields two new linear constrains on $H$ for every pair of corresponding epipoles $e_i$ and $e'_i$:

$$\begin{bmatrix} {e_i}^t & \vec{0}^t & (e'_i)_x {e_i}^t \\ \vec{0}^t & {e_i}^t & (e'_i)_y {e_i}^t \end{bmatrix}_{2\times 9} \vec{h} = 0 \tag{11}$$

where $\vec{0}^t = [0, 0, 0]$. Every pair of temporally corresponding epipoles, $e_i$ and $e'_i$, thus imposes two linear constraints on $H$. These $2n$ constraints ($i = 1, .., n$) can be added to the set of linear equations in Eq. (9) which are imposed by the homographies. Alternatively, the epipole-related constraints can be used *alone* to solve for $H$, thus avoiding the need to enforce plane-consistency on the homographies. Theoretically, four pairs of corresponding epipoles $e_i$ and $e'_i$ are sufficient.

# 4 Recovering Temporal Synchronization Between Sequences

So far we have assumed that the temporal synchronization between the two sequences is known and given. Namely, that frame $I_i$ in sequence S corresponds to frame $I'_i$ in sequence S', and therefore the transformation $T_i$ corresponds to transformation $T'_i$. Such information is often available from time stamps. However, when such synchronization is not available, we can recover it. Given two unsynchronized sequences of transformations $T_1, ...T_n$ and $T'_1, ...T'_m$, we wish to recover the unknown temporal shift $\Delta t$ between them. Let $T_i$ and $T_{i+\Delta t}$ be temporally corresponding transformations (namely, they occurred at the same time instance). Then from Eq. (5) we know that they should satisfy $eig(T_i) \parallel eig(T'_{i+\Delta t})$ (i.e., the $3 \times 1$ vectors of eigenvalues should be parallel). In other words, the similarity measure $sim(T_{t_i}, T'_{t'_i+\Delta t})$ of Eq. (6) should equal 1 (corresponding to $cos(0)$, i.e., an angle of $0°$ between the two vectors). All pairs of corresponding transformations $T_i$ and $T'_{i+\Delta t}$ must simultaneously satisfy this constraint for the correct time shift $\Delta t$. Therefore, we recover the unknown temporal time shift $\Delta t$ by maximizing the following objective function:

$$SIM(\Delta t) = \sum_i sim(T_i, T_{i+\Delta t})^2 \tag{12}$$

The maximization is currently performed by an exhaustive search over a finite range of valid time shifts $\Delta t$. To address larger temporal shifts, we apply a hierarchical search. Coarser temporal levels are constructed by composing transformations to obtain fewer transformation between more distant frames.

The objective function of Eq. (12) can be generalized to handle sequences of different frame rates, such as sequences obtained by NTSC cameras (30 frame/sec) vs. PAL cameras (25 frames/sec). The ratio between frames corresponding to equal time steps in the two sequences is $25 : 30 = 5 : 6$. Therefore, the objective function that should be maximized for an NTSC-PAL pair of sequences is:

$$SIM(\Delta t) = \sum_i sim(T^{\,5(i+1)}_{\,5i}, T'^{\,6(i+1)+\Delta t}_{\,6i+\Delta t})^2 \tag{13}$$

Where $T_i^j$ is the transformation from frame $I_i$ to frame $I_j$. In our experiments, all sequences were obtained by PAL video cameras. Therefore only the case of equal frame-rate (Eq. (12)) was experimentally verified. We found this method to be very robust. It successfully recovered the temporal shift up to *field* (sub-frame) accuracy. Sub-field accuracy may be further recovered by interpolating the values of $SIM(\Delta t)$ obtained at discrete time shifts.

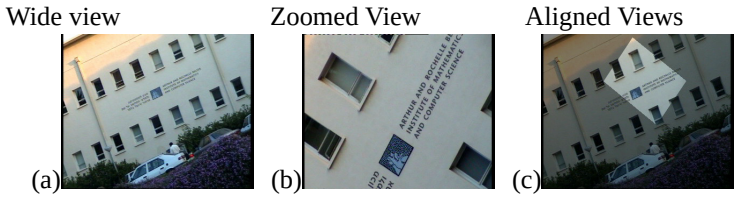



Figure 5: **Finding zoomed region.** *(a) and (b) are (automatically detected) temporally corresponding frames from the wide-FOV and the narrow-FOV sequences, respectively. (c) shows the result of spatio-temporal alignment of the two sequences, displayed by color averaging.*

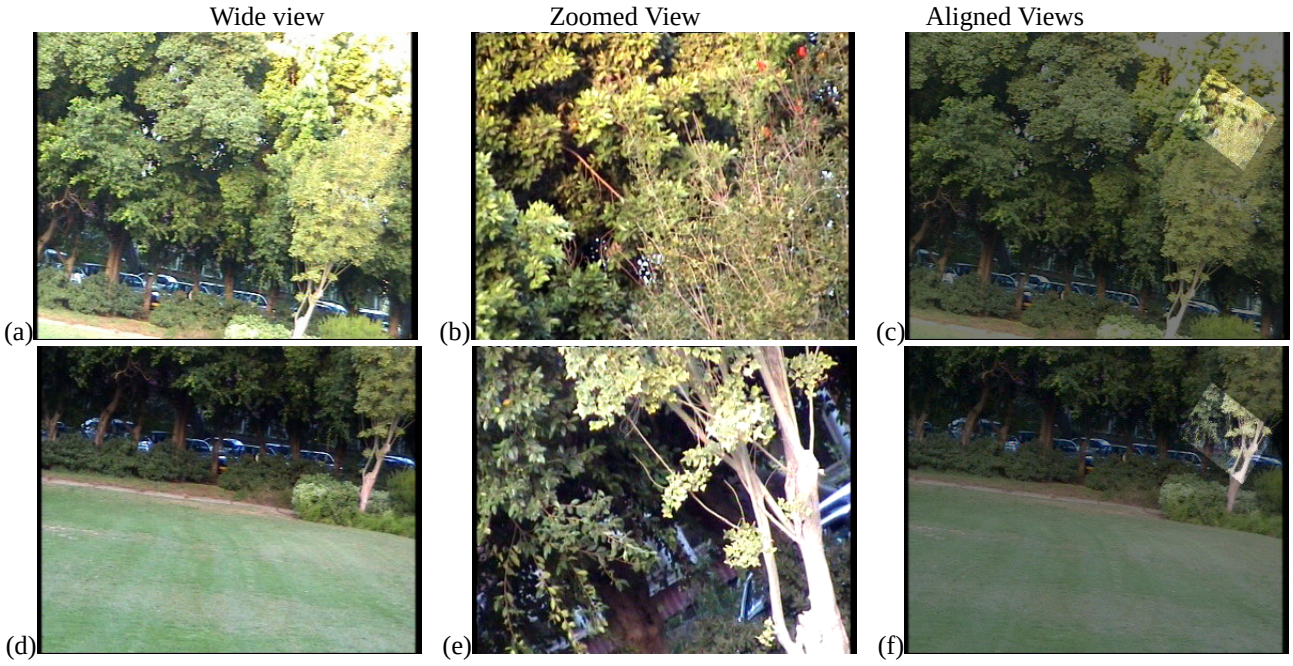



Figure 6: **Finding zoomed region.** *(a) and (d) are frames from the wide-FOV sequence. (b) and (e) are temporally corresponding frames from the narrow-FOV sequence. The correct time shift was automatically detected. (c) and (f) show super-position of the two sequences after spatio-temporal alignment, displayed by color averaging.* **For full color sequences see attached tar file.**

# 5 Applications

This section illustrates the applicability of our method to solving some real-world problems, which are particularly difficult for standard image alignment techniques. These include: (i) Alignment of non-overlapping sequences for generation of wide-screen movies from multiple narrow-screen movies (such as in IMAX films), (ii) Alignment of sequences obtained at significantly different zooms (e.g., for surveillance applications), and (iii) Alignment of multi-sensor sequences for multi-sensor fusion. We show results of applying the method to complex real-world sequences. In addition, in order to empirically quantify the accuracy of our method, we also applied it to pairs of sequences generated from a real sequence by warping it with known (ground truth) homographies. All sequences which we experimented with were captured by "of the shelf" consumer CCD cameras. The cameras were attached to each other, to minimize the distance between their centers of projections. The joint camera motion was performed manually (i.e., A person would manually hold and rotate the two attached cameras). No temporal synchronization tool was used.

The frame-to-frame input transformations within each sequence (homographies $T_1 ... T_n$ and $T'_1 ... T'_n$) were extracted using the method described in [12]. The input sequences were usually several seconds long to guaranty significant enough motion. The temporal time shift was recovered using the algorithm described in Sec. 4 up to field accuracy (This was verified against instantaneous events that were observed by both cameras).

Inaccurate frame-to-frame transformations $T_i$ are pruned out by using two outlier detection mechanisms:
(i) The transformation between successive frames within each sequence are computed in both direction. We measure the distance of the composed matrix $T_i T_i^{Reverse}$ from the identity matrix in the image space, in terms of the maximal residual misalignment of pixels.

$$Reliability(T_i) = \max_{p \in I_i} || \left( T_i T_i^{Reverse} - I \right) p || \qquad (14)$$

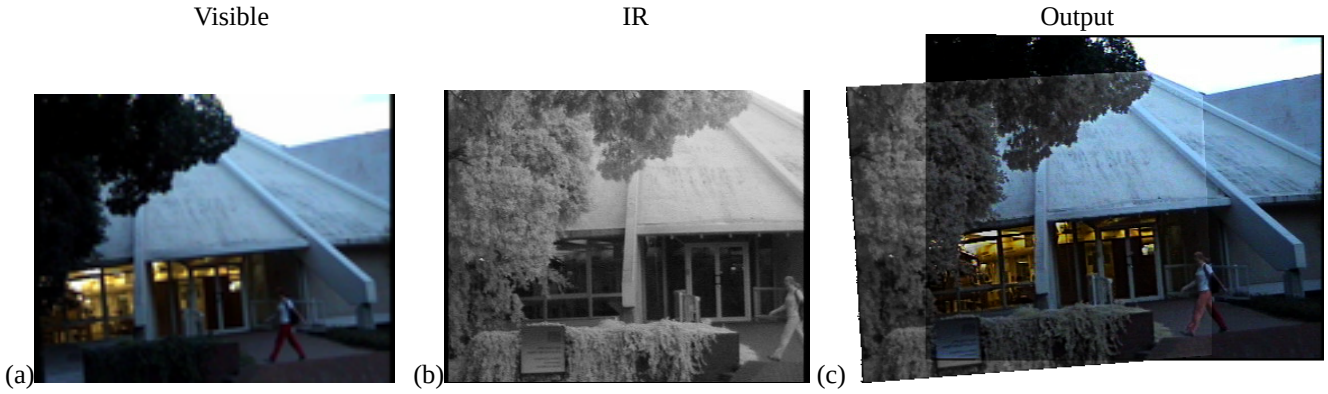



Figure 7: **Multi-sensor Alignment.** *(a) and (b) show an example of (automatically detected) temporally corresponding frames from the visible-light and IR sequences, respectively. The inside of the building is visible only in the visible-light sequence, while the IR sequence captures the details outdoors (e.g., the dark trees, the sign, the bush). (c) shows the results of fusing the two sequences after spatio-temporal alignment. The fused sequence preserves the details from both sequences. Note the high accuracy of alignment (both in time and in space) of the walking lady. For more details see text.* **For full color sequences see attached tar file.**

(ii) The similarity criterion of Eq. (6) can also be used to verify the degree of "similarity" between $T_i$ and $T'_i$. An unreliable pair of transformations can thus be pruned out. However,the first outlier criteria proved to be more powerful.

Finally, the *best* thirty or so transformations were used in the estimation of the inter-camera homography $H$ (using the algorithm described in Sec. 3.1).

## 5.1 Alignment of Non-Overlapping Sequences

Fig 3 shows an example of alignment of non-overlapping sequences. The left camera is zoomed-in and rotated relative to the right camera. The correct spatio-temporal alignment can be seen in Fig. 3.c. Note the accurate alignment of the running person both in time and in space.

Our approach to sequence alignment can be used to generate wide-screen movies from two (or more) narrow field-of-view movies (such as in IMAX movies). Such an example is shown in Fig. 4. To verify the accuracy of alignment (both in time and in space), we allowed for a very small overlap between the two sequences. However, this image region was *not* used in the estimation process, to imitate the case of truly *non-overlapping* sequences. The overlapping region was used only for display and verification purposes. Fig. 4.c shows the result of combining the two sequences (by averaging corresponding frames) after spatio-temporal alignment. Note the accurate *spatial* as well as *temporal* alignment of the soccer players in the averaged overlapping region. To see the generated natural-looking wide-screen movie, see attached tar file.

In order to empirically verify the accuracy of our method, the real video sequence of Fig. 8 was split in the middle, producing two non-overlapping sub-sequences of half-a-frame width each. The true (ground truth) homography in this case corresponds to a horizontal shift by the width of a frame ($352$ pixels). The frame-to-frame transformation ($T_1 ... T_n$ and $T'_1 ... T'_n$) were estimated separately within each sequence using [12]. The temporal shift ($\Delta t = 0$) was recovered correctly from these transformations, and the "inter-camera" homography $H$ was recovered up to a misalignment error of less than 0.7 pixel over the entire image. See table 1 for summary of results.

## 5.2 Alignment of Sequences Obtained at Different Zooms

Often in surveillance applications two cameras are used, one with a wide FOV (field-of-view) for observing large scene regions, and the other camera with a narrow FOV (zoomed-in) for detecting small objects. Matching two such images obtained at significantly different zooms is a difficult problem for standard image alignment methods, since the two images display different features which are prominent at the different resolutions. Our sequence alignment approach may be used for such scenarios. Fig. 5 shows such an example. The zoom in this case was approximately $1:3$. The result (Fig. 5.c) is displayed in the form of averaging temporally corresponding frames after alignment according to the computed homography and the computed time shift.

The same approach was further applied to a pair of sequences where the very large zoom ($\approx 1:4$) and the dense clutter in the scene make the task difficult even for the human eye. This example is shown in Fig. 6. The output is displayed by averaging temporally corresponding frames from the two sequences after spatio-temporal sequence alignment. Note the small red flowers in the zoomed view (Fig. 6.b). These cannot be seen in the corresponding low resolution wide-view frame (Fig. 6.a). Only when

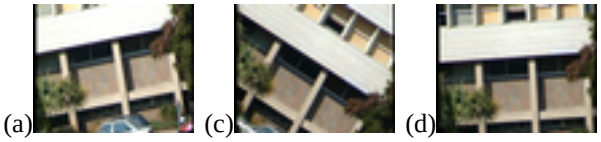



Figure 8: **The manipulated sequence.** *(a,b,c) are three frames (0,150,300) out of the original 300 frames. This sequence was used in all the quantitative experiments.*

a well-defined object (e.g., the tree stem) enters both FOV's (Figs. 6.d and 6.e), can we verify the high accuracy of the alignment.

To empirically verify the accuracy of our method in the presence of large zooms and large rotations, we ran the algorithm on following three manipulated sequences with known (ground truth) manipulations: We warped the sequence of Fig. 8 once by a zoom factor of 2, once by a zoom factor of 4, and once rotated it by $180^o$. The results are shown in table 1.

In each of these cases, the recovered homography was composed with the inverse of the ground-truth homography: $H_{true}^{-1} H_{recovored}$. Ideally, the composed homography should be the identity matrix. The errors reported in table 1 are the *maximal* residual misalignment induced by the composed homography over the entire image.

### 5.3 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 appearing in one image may not appear in the other, and visa versa. This poses a problem for image alignment methods. Our sequence alignment approach, however, does not require coherent appearance between the two sequences, and can therefore be applied to solve the problem. Fig. 7 shows an example of two such sequences, one captured by a near IR camera, while the other by a regular video (visible-light) camera. The scene was shot in twilight. In the sequence obtained by the regular video camera (Fig.7.(a)), the outdoor scene is barely visible, while the inside of the building is clearly visible. The IR camera, on the other hand, captures the outdoor scene in great detail, while the indoor part (illuminated by "cold" neon light) was invisible to the IR camera (Fig. 7.(b)). The result of the spatio-temporal alignment is illustrated by fusing temporally corresponding frames. The IR camera provides only intensity information, and was therefore fused only with the intensity (Y) component of the visible-light camera, using the image-fusion method of [3] The chrome components (I and Q) of the visible-light camera supply the color information.

For full color sequences of the results presented in this section, see attached tar file.

| **Applied Transformation** | **Recovered Transformation** | **Max Residual Misalignment** |
|---|---|---|
| Zoom factor = 2 | Zoom factor = 1.9992 | 0.4 pixels |
| Zoom factor = 4 | Zoom factor = 4.0048 | 0.4 pixels |
| Rotation by $180^o$ | Rotation by $180.00^o$ | 0.01 pixels |
| Horizontal shift of 352 pixels | Horizontal shift of 351.6 pixels | 0.7 pixels |

Table 1: **Quantitative results.** *This table summarizes the quantitative results with respect to ground truth. A real video sequence (Fig. 8) was warped ("manipulated") by a known homography, to generate a second sequence. The left column describes the type of transformation applied to the sequence, the center column describes the recovered transformation, and the right column describes the residual error between the ground-truth homography and the recovered homography (measured in maximal residual misalignment in the image space). See text for further details.*

## 6 Conclusion

This paper presents an approach for aligning two sequences (both in time and in space), even when there is no common spatial information between the sequences. This was made possible by replacing the need for "coherent appearance" (which is a fundamental requirement in standard images alignment techniques), with the requirement of "coherent temporal behavior", which is often easier to satisfy. We demonstrated applications of this approach to real-world problems, which are inherently difficult for regular image alignment techniques.

## References

[1] S. Avidan and A. Shashua. Thereading fundamaental matrices. In *European Conference on Computer Vision*, 1998.

[2] A. Bjorck. *Numerical Methodes for Least Squares Problems.* SIAM, Philadelphia, 1996.

[3] P.R. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *International Conference on Computer Vision*, 1993.

[4] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment, to appear in. In *IEEE Conference on Computer Vision and Pattern Recognition,* Hilton Head Island, South Carolina, June 2000.

[5] D. Demirdijian, A. Zisserman, and R. Horaud. Stereo autocalibration from one plane. In *European Conference on Computer Vision*, 2000.

[6] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition,* Hilton Head Island, South Carolina, June 2000.

[7] C. E.P̃earson (ed.). *Handbook of applied mathematics - Second Edition.* Van Nostrand Reinhold Company, New York, 1983.

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

[9] R. Horaud and G. Csurka. reconstruction using motions of a stereo rig. In *ICCV*, pages 96–103, 1998.

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

[11] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, Freiburg, June 1998.

[12] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, January 1994.

[13] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *Proc 12th ICPR*, 1994.

[14] C.C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, 1980.

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

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

[17] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *International Conference on Computer Vision*, pages 16–23, 1995.

[18] A. Zisserman, P.A. Beardsley, and I.D. Reid. Metric calibration of a stereo rig. In *Workshop on Representations of Visual Scenes*, pages 93–100, 1995.

# Aligning Non-Overlapping 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

December 4, 2001

[*] A shorter version of this paper appeared in ICCV 2001 [6].

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

# Contents

**1 Introduction** **4**

**2 Problem Formulation** **5**

**3 Recovering Spatial Alignment Between Sequences** **5**
3.1 Planar or Distant (2D) Scenes . . . . . 6
3.2 3D Scenes . . . . . 8

**4 Recovering Temporal Synchronization Between Sequences** **10**

**5 Applications** **10**
5.1 Alignment of Non-Overlapping Sequences . . . . . 11
5.2 Alignment of Sequences Obtained at Different Zooms . . . . . 12
5.3 Multi-Sensor Alignment . . . . . 12

**6 Analysis** **13**
6.1 Empirical Evaluation . . . . . 13
6.2 Uniqueness of Solution . . . . . 13
6.3 Numerical Stability . . . . . 15

**7 Conclusion** **16**

## Abstract

*This paper shows how two image sequences that have no spatial overlap between their fields of view can be aligned both in time and in space. Such alignment is possible when the two cameras are attached closely together and are moved jointly in space. The common motion induces "similar" changes over time within the two sequences. This correlated temporal behavior, is used to recover the spatial and temporal transformations between the two sequences. The requirement of "consistent appearance" in standard image alignment techniques is therefore replaced by "consistent temporal behavior", which is often easier to satisfy.*

*This approach to alignment can be used not only for aligning non-overlapping sequences, but also for handling other cases that are inherently difficult for standard image alignment techniques. We demonstrate applications of this approach to three real-world problems: (i) alignment of non-overlapping sequences for generating wide-screen movies, (ii) alignment of images (sequences) obtained at significantly different zooms, for surveillance applications, and, (iii) multi-sensor image alignment for multi-sensor fusion.*

**Keywords:** Spatio-temporal alignment, Temporal synchronization, Multi-sensor alignment, Alignment for wide-screen movies, Alignment across different zooms.

# 1 Introduction

The problem of image alignment (or registration) has been extensively researched, and successful approaches have been developed for solving this problem. Some of these approaches are based on matching extracted local image features, other approaches are based on directly matching image intensities. A review of some of these methods can be found in [24] and [16]. However, all these approaches share one basic assumption: that there is sufficient overlap between the two images to allow extraction of common image properties, namely, that there is sufficient "similarity" between the two images ("Similarity" of images is used here in the broadest sense. It could range from gray-level similarity, to feature similarity, to similarity of frequencies, and all the way to statistical similarity such as mutual information [26]).

In this paper the following question is addressed: *Can two images be aligned when there is very little similarity between them, or even more extremely, when there is no spatial overlap at all between the two images?* When dealing with individual images, the answer tends to be "No". However, this is not the case when dealing with image sequences. An image sequence contains much more information than any individual frame does. In particular, temporal changes (such as dynamic changes in the scene, or the induced image motion) are encoded *between* video frames, but do not appear in any individual frame. Such information can form a powerful cue for alignment of two (or more) sequences. Caspi and Irani [5] and Stein [23] have illustrated an applicability of such an approach for aligning two sequences based on common dynamic scene information. However, they assumed that the same temporal changes in the scene (e.g., moving objects) are visible to both video cameras, leading to the requirement that there must be significant overlap in the FOVs (fields-of-view) of the two cameras.

In this paper we show that when two cameras are attached closely to each other (so that their centers of projections are very close), and move jointly in space, then the induced frame-to-frame transformations *within* each sequence have correlated behavior *across* the two sequences. This is true even when the sequences have no spatial overlap. This correlated temporal behavior is used to recover both the spatial and temporal transformations between the two sequences.

Unlike carefully calibrated stereo-rigs [22], our approach does not require any prior internal or external camera calibration, nor any sophisticated hardware. Our approach bears resemblance to the approaches suggested by [7, 14, 27] for auto-calibration of stereo-rigs. But unlike these methods, we do not require that the two cameras observe and match the same scene features, nor that their FOVs will overlap.

The need for "consistent appearance", which is a fundamental assumption in image alignment or calibration methods, is replaced here with the requirement of "consistent temporal behavior". Consistent temporal behavior is often easier to satisfy (e.g., by moving the two cameras jointly in space). A similar idea was used for "hand-eye calibration" in robotics research (e.g., [25, 15]).

Our approach is useful not only in the case of non-overlapping sequences, but also in other cases where there is very little common appearance information between images, and are therefore inherently difficult for standard image alignment techniques. This gives rise to a variety of real-world applications, including: (i) Multi-sensor alignment for image fusion. This requires accurate alignment of images (sequences) obtained by sensors of different sensing modalities (such as Infra-Red and visible light). Such images differ significantly in their appearance due to different sensor properties [26]. (ii) Alignment of images (sequences) obtained at different zooms. The problem here is that different image features are prominent at different image resolutions [8]. Alignment of a wide-FOV sequence with a narrow-FOV sequence is useful for detecting small zoomed-in objects in (or outside) a zoomed-out view of the scene. This can be useful in surveillance applications. (iii) Generation of wide-screen movies from multiple non-overlapping narrow FOV movies (such as in IMAX movies).

Our approach can handle such cases. Results are demonstrated in the paper on complex real-world sequences, as well as on manipulated sequences with ground truth.

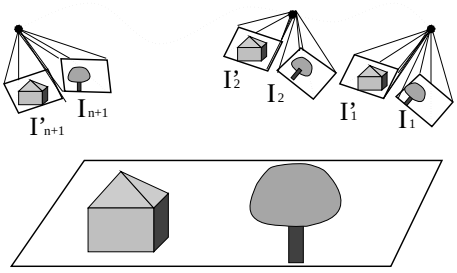



Figure 1: *Two video cameras are attached to each other, so that they have the same center of projection, but non-overlapping fields-of-view. The two cameras are moved jointly in space, producing two separate video sequences* $I_1, ..., I_{n+1}$ *and* $I'_1, ..., I'_{n+1}$.

## 2 Problem Formulation

We examine the case when two video cameras having (approximately) the same center of projection but different 3D orientation, move jointly in space (see Fig. 1). The fields of view of the two cameras do not necessarily overlap. The internal parameters of the two cameras are different and unknown, but fixed along the sequences. The external parameters relating the two cameras (i.e., the relative 3D orientation) are also unknown but fixed. Let $S = I_1, ...I_{n+1}$ and $S' = I'_1, ..., I'_{m+1}$ be the two sequences of images recorded by the two cameras[1]. When temporal synchronization (e.g., time stamps) is not available, then $I_i$ and $I'_i$ may not be corresponding frames in time. Our goal is to recover the transformation that aligns the two sequences both in time and in space. Note the term "alignment" here has a broader meaning than the usual one, as the sequences may not overlap in space, and may not be synchronized in time. Here we refer to alignment as displaying one sequence in the spatial coordinate system of the other sequence, and at the correct time shift, as if obtained by the other camera.

When the two cameras have the same center of projection (and differ only in their 3D orientation and their internal calibration parameters), then a simple fixed homography $H$ (a 2D projective transformation) describes the *spatial* transformation between *temporally corresponding pairs of frames* across the two sequences [12].

If there were enough common features (e.g., $p$ and $p'$) between temporally corresponding frames (e.g., $I_i$ and $I'_i$), then it would be easy to recover the inter-camera homography $H$, as each such pair of corresponding image points would provide linear constrains on $H$: $p' \cong Hp$. This, in fact, is how most image alignment techniques work [12]. However, this is not the case here. The two sequence do not share common features, because there is no spatial overlap between the two sequences. Instead, the homography $H$ is recovered from the induced frame-to-frame transformations *within* each sequence.

Let $T_1, ...T_n$ and $T'_1, ...T'_m$ be the sequences of frame-to-frame transformations within the video sequences $S$ and $S'$, respectively. $T_i$ is the transformation relating frame $I_i$ to $I_{i+1}$. These transformations can be either 2D parametric transformations (e.g., homographies or affine transformations) or 3D transformations/relations (e.g., fundamental matrices). We next show how we can recover the spatial transformation $H$ and the temporal shift $\Delta t$ between the two video sequences directly from the two sequences of transformations $T_1, ...T_n$ and $T'_1, ...T'_m$. The problem formulated above is illustrated in Fig. 2.

## 3 Recovering Spatial Alignment Between Sequences

Let us first assume that the temporal synchronization is known. Such information is often available (e.g., from time stamps encoded in each of the two sequences). Sec. 4 shows how we can recover the temporal shift between the

[1]The subscript $i$ is used to represent the frame time index, and the superscript prime is used to distinguish between the two sequences $S$ and $S'$.

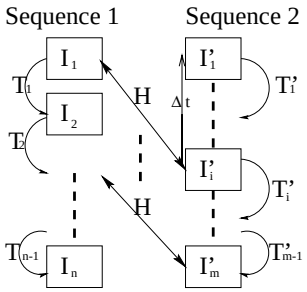



Figure 2: **Problem formulation.** *The two sequences are spatially related by a fixed but unknown inter-camera homography H, and temporally related by a fixed and unknown time shift $\Delta t$. Given the frame-to-frame transformations $T_1, ..., T_n$ and $T'_1, ..., T'_m$, we want to recover $H$ and $\Delta t$.*

two sequences when that information is not available. Therefore, without loss of generality, it is assumed that $I_i$ and $I'_i$ are corresponding frames in time in sequences $S$ and $S'$, respectively. Two cases are examined: (i) The case when the scene is planar or distant from the cameras. We refer to these scenes as "2D scenes". In this case the frame-to-frame transformations $T_i$ can be modeled by homographies (Sec. 3.1). (ii) The case of a non-planar scene. We refer to these scenes as "3D scenes". In this case the frame-to-frame relations can be modeled by fundamental matrices (Sec. 3.2).

### 3.1 Planar or Distant (2D) Scenes

When the scene is planar or distant from the cameras, or when the joint 3D translations of the two cameras are negligible relative to the distance of the scene, then the induced image motions within each sequence (i.e., $T_1, ...T_n$ and $T'_1, ...T'_n$) can be described by 2D parametric transformations [12]. $T_i$ thus denotes the homography between frame $I_i$ and $I_{i+1}$, represented by $3 \times 3$ non-singular matrices. We next show that temporally corresponding *transformations* $T_i$ and $T'_i$ are related by the same fixed inter-camera homography $H$ (which relates frames $I_i$ and $I'_i$).

Let $P$ be a 3D point in the planar (or the remote) scene. Denote by $p_i$ and $p'_i$ its image coordinates in frames $I_i$ and $I'_i$, respectively (the point $P$ need not be visible in the two frames, i.e., $P$ need not be within the FOV of the cameras). Let $p_{i+1}$ and $p'_{i+1}$ be its image coordinates in frames $I_{i+1}$ and $I'_{i+1}$, respectively. Then, $p_{i+1} \cong T_i p_i$ and $p'_{i+1} \cong T'_i p'_i$. Because the coordinates of the video sequences $S$ and $S'$ are related by a fixed homography $H$ , then: $p' \cong Hp$ and $p'_{i+1} \cong Hp_{i+1}$. Therefore:

$$HT_i p_i \cong Hp_{i+1} \cong p'_{i+1} \cong T'_i p'_i \cong T'_i Hp_i \tag{1}$$

Each $p_i$ could theoretically have a different scalar associated with the equality in Eq. (1). However, it is easy to show that because the relation in Eq. (1) holds for *all* points $p_i$, therefore all these scalars are equal, and hence:

$$HT_i \cong T'_i H. \tag{2}$$

Because $H$ is non-singular we may write $T'_i \cong HT_i H^{-1}$, or

$$T'_i = s_i HT_i H^{-1} \tag{3}$$

where $s_i$ is a (frame-dependent) scale factor. Eq. (3) is true for all frames, i.e., for any pair of corresponding transformations $T_i$ and $T'_i$ ($i = 1..n$) there exists a scalar $s_i$ such that $T'_i = s_i HT_i H^{-1}$. It shows that there is a **similarity relation**[2] (or a "conjugacy relation") between the two matrices $T_i$ and $T'_i$ (up to a scale factor). A similar observation was made for case of hand-eye calibration (e.g., [25, 15]), and for auto-calibration of a stereo-rig (e.g. [27]).

[2]A matrix $A$ is said to be "similar" to a matrix $B$ if there exists an invertible matrix $M$ such that $A = MBM^{-1}$ (see [9]). The term *"conjugate matrices"* is also often used.

Denote by $eig(A) = [\lambda_1, \lambda_2, \lambda_3]^t$ a $3 \times 1$ vector containing the eigenvalues of a $3 \times 3$ matrix $A$ (in decreasing order). Then it is known ([9] pp. 898.) that: (i) If $A$ and $B$ are similar (conjugate) matrices, then they have the same eigenvalues: $eig(A) = eig(B)$, and, (ii) The eigenvalues of a scaled matrix are scaled: $eig(sA) = s(eig(A))$. Using these two facts and Eq. (3) we obtain:

$$eig(T_i') = s_i \; eig(T_i) \tag{4}$$

where $s_i$ is the scale factor defined by Eq. (3). Eq. (4) implies that the two vectors $eig(T_i)$ and $eig(T_i')$ are "parallel". This gives rise to a measure of similarity between two matrices $T_i$ and $T_i'$:

$$sim(T_i, T_i') = \frac{eig(T_i)^t \; eig(T_i')}{||eig(T_i)|| \; ||eig(T_i')||}, \tag{5}$$

where $||\cdot||$ is the vector norm. For real valued eigenvalues, Eq. (5) provides the cosine of the angle between the two vectors $eig(T_i)$ and $eig(T_i')$. This property will be used later for obtaining the temporal synchronization between the two sequences (Sec. 4). This measure is also used for outlier rejection of bad frame-to-frame transformation pairs, $T_i$ and $T_i'$ (Appendix A). The remainder of this section explains how the fixed inter-camera homography H is recovered from the list of frame-to-frame transformations $T_1, ..T_n$ and $T_1', .., T_n'$.

For each pair of temporally corresponding transformations $T_i$ and $T_i'$ in sequences $S$ and $S'$, we first compute their eigenvalues $eig(T_i)$ and $eig(T_i')$. The scale factor $s_i$ which relates them is then estimated from Eq. (4) using least squares minimization (three equations, one unknown)[3]. Once $s_i$ is estimated, Eq. (3) (or Eq. (2)) can be rewritten as:

$$s_i H T_i - T_i' H = 0 \tag{6}$$

Eq. (6) is linear in the unknown components of $H$. Rearranging the components of $H$ in a $9 \times 1$ column vector $\vec{h} = [H_{11}H_{12}H_{13}H_{21}H_{22}H_{23}H_{31}H_{32}H_{33}]^t$, Eq. (6) can be rewritten as a set of linear equations in $\vec{h}$:

$$M_i \vec{h} = \vec{0} \tag{7}$$

where $M_i$ is a $9 \times 9$ matrix defined by $T_i, T_i'$ and $s_i$:

$$M_i = \left[\begin{array}{c|c|c} s_i {T_i}^t - T_{i_{11}}' I & -T_{i_{12}}' I & -T_{i_{13}}' I \\ \hline -T_{i_{21}}' I & s_i T^t - T_{i_{22}}' I & -T_{i_{23}}' I \\ \hline -T_{i_{31}}' I & -T_{i_{32}}' I & s_i T^t - T_{i_{33}}' I \end{array}\right]_{9\times 9}$$

where $I$ is the $3 \times 3$ identity matrix. Eq. (7) implies that each pair of corresponding transformations $T_i$ and $T_i'$ contributes 9 linear constrains in the unknown homography $H$ (i.e., $\vec{h}$), out of which at most 6 constraints are linearly independent (see Sec. 6). Therefore, in theory, at least two such pairs of independent transformations are needed to uniquely determine the homography $H$ (up to a scale factor). In practice, we use all available constraints from all pairs of transformations to compute $H$. The constraints from all the transformations $T_1, .., T_n$ and $T_1', .., T_n'$ can be combined into a single set of linear equations in $\vec{h}$:

$$A\vec{h} = \vec{0} \tag{8}$$

where A is a $9n \times 9$ matrix: $A = \left[\begin{array}{c} M_1 \\ \vdots \\ M_n \end{array}\right]$. Eq. (8) is a homogeneous set of linear equations in $\vec{h}$, that can be solved in a variety of ways [3]. In particular, $\vec{h}$ may be recovered up to scale by computing the eigenvector which corresponds to the smallest eigenvalue of the matrix $A^tA$.

[3] Alternatively, the input homographies can be normalized to have determinant equal to 1, to avoid the need to compute $s_i$.


Figure 3: **Alignment of non-overlapping sequences.** *(a) and (b) are temporally corresponding frames from sequences $S$ and $S'$. The correct time shift was automatically detected. (c) shows one frame in the combined sequence after spatio-temporal alignment. Note the accuracy of the spatial and temporal alignment of the running person.* **For full sequences see www.wisdom.weizmann.ac.il/NonOverlappingSeqs.**

### 3.2 3D Scenes

When the scene is neither planar nor distant, the relation between two consecutive frames of an uncalibrated camera is described by the fundamental matrix [12]. In this case the input to our algorithm is two sequences of fundamental matrices between successive frames, denoted by $F_1, ...F_n$ and $F'_1, ...F'_n$. Namely, if $p_i \in I_i$ and $p_{i+1} \in I_{i+1}$ are corresponding image points, then: $p^t_{i+1} F_i p_i = 0$. Although the relations *within* each sequence are characterized by fundamental matrices, the inter-camera transformation remains a homography $H$. This is because the two cameras still share the same center of projection (Sec. 2).

Each fundamental matrix $F_i$ can be decomposed into a homography + epipole [12] as follows:

$$F_i = [e_i]_\times T_i$$

where $e_i$ is the epipole relating frames $I_i$ and $I_{i+1}$, the matrix $T_i$ is the induced homography from $I_i$ to $I_{i+1}$ via any plane (real or virtual). $[\cdot]_\times$ is the cross product matrix ($[v]_\times \vec{w} = \vec{v} \times \vec{w}$).

The homographies, $T_1, ..., T_n$ and $T'_1, ..., T'_n$, and the epipoles $e_1, ..., e_n$ and $e'_1, ..., e'_n$, impose separate constraints on the inter-camera homography $H$. These constraints can be used separately or jointly to recover $H$.
**(i) Homography-based constraints:** The homographies $T_1, .., T_n$ and $T'_1, .., T'_n$ (extracted from the fundamental matrices $F_1, .., F_n$ and $F'_1, .., F'_n$, respectively), may correspond to different 3D planes. In order to apply the algorithm of Sec. 3.1 using these homographies, we need to impose plane-consistency across the two sequences (to guarantee that temporally corresponding homographies correspond to the same plane in the 3D world). One possible way for imposing plane-consistency across (and within) the two sequences is by using the "Plane+Parallax"

Figure 4: **Wide-screen movies generation** *(a) and (b) are temporally corresponding frames from sequences $S$ and $S'$. The correct time shift was automatically detected. (c) shows one frame in the combined sequence. Corresponding video frames were averaged after spatio-temporal alignment. The small overlapping area was not used in the estimation process, but only for verification (see text). Note the accuracy of the spatial and temporal alignment of the soccer player in the overlapping region.* **For full sequences see www.wisdom.weizmann.ac.il/NonOverlappingSeqs.**

approach [19, 17, 21, 20]. However, this approach requires that a real physical planar surface be visible in *all* video frames. Alternatively, the "threading" method of [1] or other methods for computing consistent set of camera matrices (e.g., [2]), can impose plane-consistency within each sequence, even if no real physical plane is visible in any of the frames. Plane consistency *across* the two sequences can be obtained, e.g., if [1] is initiated at frames which are known to simultaneously view the same real plane in both sequences. This can be done even if the two cameras see different portions of the plane (allowing for non-overlapping FOVs), and do not see that plane at any of the other frames. This approach is therefore less restrictive than the Plane+Parallax approach.

**(ii) Epipole-based constraints:** The fundamental matrices $F_1..F_n$ and $F'_1..F'_n$ also provide a list of epipoles $e_1, ..., e_n$ and $e'_1, ..., e'_n$. These epipoles are uniquely defined (there is no issue of plane consistency here).

Since the two cameras have the same center of projection, then for any frame $i$: $e'_i \cong He_i$, or more specifically:

$$(e'_i)_x = \frac{[h_1 h_2 h_3]\, e_i}{[h_7 h_8 h_9]\, e_i} \quad (e'_i)_y = \frac{[h_4 h_5 h_6]\, e_i}{[h_7 h_8 h_9]\, e_i} \tag{9}$$

Multiplying by the dominator and rearranging terms yields two new linear constrains on $H$ for every pair of corresponding epipoles $e_i$ and $e'_i$:

$$\begin{bmatrix} {e_i}^t & \vec{0}^t & (e'_i)_x {e_i}^t \\ \vec{0}^t & {e_i}^t & (e'_i)_y {e_i}^t \end{bmatrix}_{2\times 9} \vec{h} = 0 \tag{10}$$

where $\vec{0}^t = [0,0,0]$. Every pair of temporally corresponding epipoles, $e_i$ and $e'_i$, thus imposes two linear constraints on $H$. These $2n$ constraints ($i = 1,..,n$) can be added to the set of linear equations in Eq. (8) which are imposed by the homographies. Alternatively, the epipole-related constraints can be used *alone* to solve for $H$, thus avoiding the need to enforce plane-consistency on the homographies. Theoretically, four pairs of corresponding epipoles $e_i$ and $e'_i$ in general position (no 3 on the same line) are sufficient.

# 4 Recovering Temporal Synchronization Between Sequences

So far we have assumed that the temporal synchronization between the two sequences is known and given. Namely, that frame $I_i$ in sequence $S$ corresponds to frame $I'_i$ in sequence $S'$, and therefore the transformation $T_i$ corresponds to transformation $T'_i$. Such information is often available from time stamps. However, when such synchronization is not available, we can recover it. Given two unsynchronized sequences of transformations $T_1, ...T_n$ and $T'_1, ...T'_m$, we wish to recover the unknown temporal shift $\Delta t$ between them. Let $T_i$ and $T'_{i+\Delta t}$ be temporally corresponding transformations (namely, they occurred at the same time instance). Then from Eq. (4) we know that they should satisfy $eig(T_i) \parallel eig(T'_{i+\Delta t})$ (i.e., the $3 \times 1$ vectors of eigenvalues should be parallel). In other words, the similarity measure $sim(T_{t_i}, T'_{t'_i+\Delta t})$ of Eq. (5) should equal 1 (corresponding to $cos(0)$, i.e., an angle of $0^\circ$ between the two vectors). All pairs of corresponding transformations $T_i$ and $T'_{i+\Delta t}$ must simultaneously satisfy this constraint for the correct time shift $\Delta t$. Therefore, we recover the unknown temporal time shift $\Delta t$ by maximizing the following objective function:

$$SIM(\Delta t) = \sum_i sim(T_i, T_{i+\Delta t})^2 \tag{11}$$

The maximization is currently performed by an exhaustive search over a finite range of valid time shifts $\Delta t$. To address larger temporal shifts, we apply a hierarchical search. Coarser temporal levels are constructed by composing transformations to obtain fewer transformation between more distant frames.

The objective function of Eq. (11) can be generalized to handle sequences of different frame rates, such as sequences obtained by NTSC cameras (30 frame/sec) vs. PAL cameras (25 frames/sec). The ratio between frames corresponding to equal time steps in the two sequences is $25 : 30 = 5 : 6$. Therefore, the objective function that should be maximized for an NTSC-PAL pair of sequences is:

$$SIM(\Delta t) = \sum_i sim(T_{5i}^{5(i+1)}, T'^{6(i+1)+\Delta t}_{6i+\Delta t})^2 \tag{12}$$

Where $T_i^j$ is the transformation from frame $I_i$ to frame $I_j$. In our experiments, all sequences were obtained by PAL video cameras. Therefore only the case of equal frame-rate (Eq. (11)) was experimentally verified. We found this method to be very robust. It successfully recovered the temporal shift up to *field* (half-frame) accuracy. Sub-field accuracy may be further recovered by interpolating the values of $SIM(\Delta t)$ obtained at discrete time shifts.

# 5 Applications

This section illustrates the applicability of our method to solving some real-world problems, which are particularly difficult for standard image alignment techniques. These include: (i) Alignment of non-overlapping sequences for generation of wide-screen movies from multiple narrow-screen movies (such as in IMAX films), (ii) Alignment of sequences obtained at significantly different zooms (e.g., for surveillance applications), and (iii) Alignment of multi-sensor sequences for multi-sensor fusion. We show results of applying the method to complex real-world sequences. All sequences which we experimented with, were captured by "off-the-shelf" consumer CCD cameras. The cameras were attached to each other to minimize the distance between their centers of projections. The joint

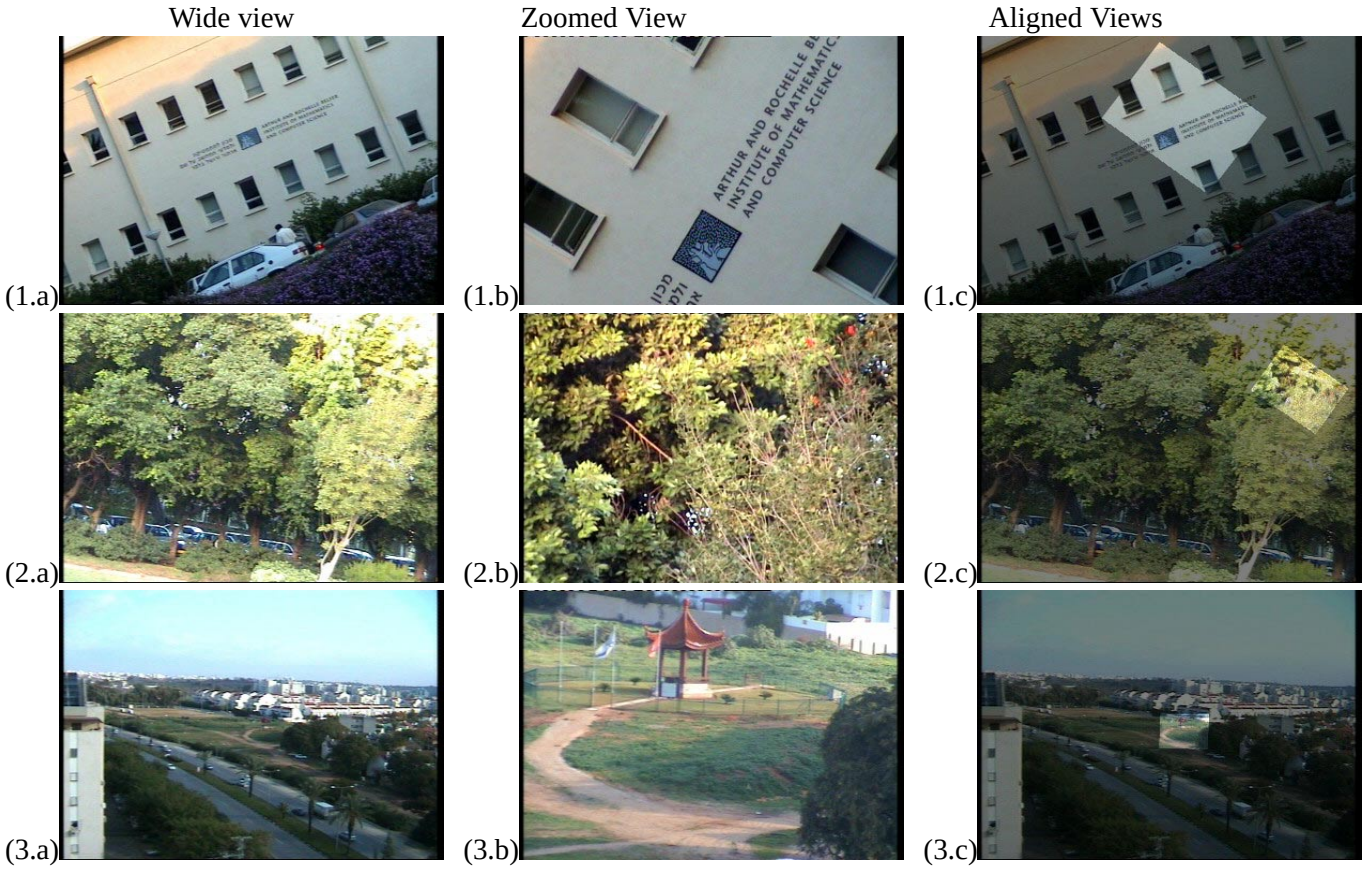



Figure 5: **Finding zoomed region.** *This figure displays three different examples (one at each row), each one with different zoom factor. The left column (1.a, 2.a, 3.a) display one frame from each of the three wide-FOV sequences. The temporally corresponding frames from the corresponding narrow-FOV sequences are displayed in the center column (1.b, 2.b, 3.b). The correct time shift was automatically detected for each pair of narrow/wide FOV sequences. Each image on the right column shows super-position of corresponding frames of the two sequences after spatio-temporal alignment, displayed by color averaging (1.c, 2.c, 3.c).* **For full sequences see www.wisdom.weizmann.ac.il/NonOverlappingSeqs.**

camera motion was performed manually (i.e., a person would manually hold and rotate the two attached cameras). No temporal synchronization tool was used.

The frame-to-frame input transformations within each sequence (homographies $T_1, ..., T_n$ and $T'_1, ..., T'_n$) were extracted using the method described in [18]. Inaccurate frame-to-frame transformations $T_i$ are pruned out by using two outlier detection mechanisms (see Appendix A). The input sequences were usually several seconds long to guaranty significant enough motion. The temporal time shift was recovered using the algorithm described in Sec. 4 up to field accuracy. Finally, the *best* thirty or so transformations were used in the estimation of the inter-camera homography $H$ (using the algorithm described in Sec. 3.1).

### 5.1 Alignment of Non-Overlapping Sequences

Fig. 3 shows an example of alignment of non-overlapping sequences. The left camera is zoomed-in and rotated relative to the right camera. The correct spatio-temporal alignment can be seen in Fig. 3.c. Note the accurate alignment of the running person both in time and in space.

Our approach to sequence alignment can be used to generate wide-screen movies from two (or more) narrow

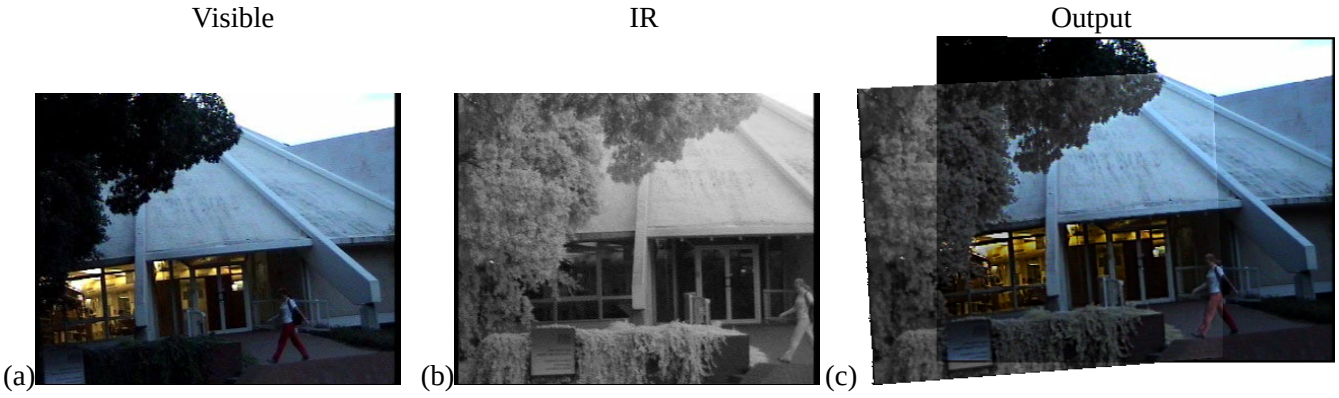



Figure 6: **Multi-sensor Alignment.** *(a) and (b) are temporally corresponding frames from the visible-light and near-IR sequences, respectively (the temporal alignment was automatically detected). The inside of the building is visible only in the visible-light sequence, while the IR sequence captures the details outdoors (e.g., the dark trees, the sign, the bush). (c) shows the results of fusing the two sequences after spatio-temporal alignment. The fused sequence preserves the details from both sequences. Note the high accuracy of alignment (both in time and in space) of the walking lady. For more details see text.* **For full sequences see www.wisdom.weizmann.ac.il/NonOverlappingSeqs.**

field-of-view movies (such as in IMAX movies). Such an example is shown in Fig. 4. To verify the accuracy of alignment (both in time and in space), we allowed for a very small overlap between the two sequences. However, this image region was *not* used in the estimation process, to imitate the case of truly *non-overlapping* sequences. The overlapping region was used only for display and verification purposes. Fig. 4.c shows the result of combining the two sequences (by averaging corresponding frames) after spatio-temporal alignment. Note the accurate spatial as well as temporal alignment of the soccer player in the averaged overlapping region.

### 5.2 Alignment of Sequences Obtained at Different Zooms

Often in surveillance applications two cameras are used, one with a wide FOV (field-of-view) for observing large scene regions, and the other camera with a narrow FOV (zoomed-in) for detecting small objects. Matching two such images obtained at significantly different zooms is a difficult problem for standard image alignment methods, since the two images display different features which are prominent at the different resolutions. Our sequence alignment approach may be used for such scenarios. Fig. 5 shows three such examples. The results of the spatio-temporal alignment (right column of Fig. 5) are displayed in the form of averaging temporally corresponding frames after alignment according to the computed homography and the computed time shift. In the first example (top row of Fig. 5) the zoom difference between the two cameras was approximately 1:3. In the second example (second row) it was $\approx$1:4, and in the third example (bottom row) it was $\approx$1:8. Note the small red flowers in the zoomed view (Fig. 5.2.b), that can barely be seen in the corresponding low resolution wide-view frame (Fig. 5.2.a). The same holds for the Pagoda in Fig. 5.3.b

### 5.3 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 appearing in one image may not appear in the other, and vice versa. This poses a problem for image alignment methods. Our sequence alignment approach, however, does not require consistent appearance between the two sequences, and can therefore be applied to solve the problem. Fig. 6 shows an example of two such sequences, one captured by a near IR camera, while the other by a regular video (visible-light) camera. The scene was shot in twilight. In the sequence obtained by the regular video camera (Fig.6.(a)), the outdoor scene is barely

(a)  (b)  (c) 

Figure 7: **The sequence used for empirical evaluation.** *(a,b,c) are three frames (0,150,300) out of the original 300 frames. This sequence was used as the base sequence for the quantitative experiments summarized in Table 1.*

visible, while the inside of the building is clearly visible. The IR camera, on the other hand, captures the outdoor scene in great detail, while the indoor part (illuminated by "cold" neon light) was invisible to the IR camera (Fig. 6.(b)). The result of the spatio-temporal alignment is illustrated by fusing temporally corresponding frames. The IR camera provides only intensity information, and was therefore fused only with the intensity (Y) component of the visible-light camera (using the image-fusion method of [4]). The chrome components (I and Q) of the visible-light camera supply the color information.

The reader is encouraged to view color sequences at www.wisdom.weizmann.ac.il/NonOverlappingSeqs.

# 6 Analysis

In this section we evaluated the effectiveness and stability of the presented approach empirically (Sec. 6.1), theoretically (Sec. 6.2) and numerically (Sec. 6.3).

## 6.1 Empirical Evaluation

In order to empirically verify the accuracy of our method, we took a real video sequence (see Fig. 7) and generated from it pairs of sequences with known (ground truth) spatial transformation $H$ and temporal shift $\Delta t$. We then applied our algorithm and compared the recovered $H$ and $\Delta t$ with the ground truth.

For the case of non overlapping sequences, the real sequence of Fig. 7 was split in the middle, producing two non-overlapping sub-sequences of half-a-frame width each. The true (ground truth) homography $H$ therefore corresponds to a horizontal shift by the width of a halved frame ($352$ pixels), and $\Delta t$ in this case is 0. The "inter-camera" homography $H$ was recovered up to a misalignment error of less than 0.7 pixel over the entire image. The temporal shift ($\Delta t = 0$) was recovered accurately from the frame-to-frame transformations.

To empirically verify the accuracy of our method in the presence of large zooms and large rotations, we ran the algorithm on following three manipulated sequences with known (ground truth) manipulations: We warped the sequence of Fig. 7 (once by a zoom factor of $2$, once by a zoom factor of $4$, and once rotated it by $180^o$) to generate the second sequence.

The results are summarized in Table 1. The reported residual misalignment was measured as follows: The recovered homography was composed with the inverse of the ground-truth homography: $H_{true}^{-1} H_{recovored}$. Ideally, the composed homography should be the identity matrix. The errors reported in Table 1 are the *maximal* residual misalignment induced by the composed homography over the entire image. In all the cases the correct $\Delta t$ was recovered (not shown in the table).

## 6.2 Uniqueness of Solution

This section studies how many pairs of corresponding transformations $T_i$ and $T_i'$ are required in order to uniquely resolve the inter-camera homography $H$. To do so we examine the number of constraints imposed on H by a single

| Applied Transformation | Recovered Transformation | Max Residual Misalignment |
|---|---|---|
| Horizontal shift of 352 pixels | Horizontal shift of 351.6 pixels | 0.7 pixels |
| Zoom factor = 2 | Zoom factor = 1.9992 | 0.4 pixels |
| Zoom factor = 4 | Zoom factor = 4.0048 | 0.4 pixels |
| Rotation by $180^o$ | Rotation by $180.00^o$ | 0.01 pixels |

Table 1: **Quantitative results.** *This table summarizes the quantitative results with respect to ground truth. Each row corresponds to one experiment. In each experiment a real video sequence (Fig. 7) was warped ("manipulated") by a known homography, to generate a second sequence. The left column describes the type of spatial transformation applied to the sequence, the center column describes the recovered transformation, and the right column describes the residual error between the ground-truth homography and the recovered homography (measured in maximal residual misalignment in the image space). In all 4 cases the correct temporal shift was recovered accurately. See text for further details.*

pair of transformations via the similarity equation Eq. (3). Since we can extract the scale factor $s_i$ directly from $T_i$ and $T_i'$ (see Sec. 3.1) we can omit the scale factor $s_i$ and study the following question: How many constraints does an equation of the form $G = HBH^{-1}$ impose on H? (e.g., $B = T_i$ and $G = T_i'$)[4].

The following notations are used: Denote by $\lambda_1, \lambda_2\, \lambda_3$ the eigenvalues of the matrix $B$ in decreasing order ($|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$). Denote by $\vec{\mathbf{u}}_{b_1}, \vec{\mathbf{u}}_{b_2}, \vec{\mathbf{u}}_{b_3}$ the corresponding eigenvectors with unit norm ($||\vec{\mathbf{u}}_{b_1}|| = ||\vec{\mathbf{u}}_{b_2}|| = ||\vec{\mathbf{u}}_{b_3}|| = 1$). Denote by $r_j$ the *algebraic multiplicity*[5] of the eigenvalue $\lambda_j$, and denote by $V_j = \{\vec{\mathbf{v}} \in R^n : B\vec{\mathbf{v}} = \lambda_j \vec{\mathbf{v}}\}$ the corresponding *eigen subspace*.

**Basic Constraints:**
Similar (conjugate) matrices (e.g., $B$ and $G$) have the same eigenvalues but different eigenvectors. Their eigenvectors are related by $H$. If $\mathbf{u_b}$ is an eigenvector of $B$ with corresponding eigenvalue $\lambda$, then $H\,\mathbf{u_b}$ is an eigenvector of $G$ with the same eigenvalue $\lambda$: $G(H\,\mathbf{u_b}) = \lambda(H\,\mathbf{u_b})$. The same holds for eigen subspaces. If $V$ is an eigen subspace of $B$ corresponding to an eigenvalue $\lambda$, then $H(V)$ is an eigen subspace of $G$ with the same eigenvalue $\lambda$. We investigate the number of constraints imposed on $H$ by $B$ and $G$ according to the dimensionality of their eigen subspaces. Let $V$ be the eigen subspace corresponding to an eigenvector $\mathbf{u_b}$ of $B$. We investigate three possible cases, one for each possible dimensionality of $V$, i.e., $dim(V) = 1, 2, 3$.
Case I: $dim(V) = 1$. This case mostly occurs when all three eigenvalues are distinct, but can also occur if some eigenvalues have algebraic multiplicity two or even three. In all these cases, $V$ is spanned by the single eigenvector $\mathbf{u_b}$. Similarly $H(V)$ is spanned by the eigenvector $\mathbf{u_g}$ of G. Therefore:

$$H\,\mathbf{u_b} = \alpha\,\mathbf{u_g} \tag{13}$$

with an unknown scale factor $\alpha$. Eq. (13) provides 3 linear equations in $H$ and one new unknown $\alpha$, thus in total it provides two new linearly independent constraints on $H$.
Case II: $dim(V) = 2$. This occurs in one of the following two cases: (a) when there exists an eigenvalue with algebraic multiplicity two, or (b) when there is only one eigenvalue with algebraic multiplicity three, but the eigen subspace spanned by all eigenvectors has dimensionality of two[6]. When $dim(V) = 2$ then two eigenvectors span $V$ (w.l.o.g., $\mathbf{u_{b1}}$ and $\mathbf{u_{b2}}$). Then every linear combination of $\mathbf{u_{b1}}$ and $\mathbf{u_{b2}}$ is also an eigenvector of $B$ with the same

[4] A general analysis of matrix equations of the form $GH = HB$ may be found in [10].

[5] If $\lambda_1 \neq \lambda_2 \neq \lambda_3$ then the algebraic multiplicity of all eigenvalues is 1 ($r_j = 1$). If $\lambda_1 = \lambda_2 \neq \lambda_3$ then the algebraic multiplicity of $\lambda_1$ and $\lambda_2$ is 2, and the algebraic multiplicity of $\lambda_3$ is 1 ($r_1 = r_2 = 2$ and $r_3 = 1$). If $\lambda_1 = \lambda_2 = \lambda_3$ then the algebraic multiplicity of $\lambda_1$,$\lambda_2$,and $\lambda_2$ is 3 ($r_1 = r_2 = r_3 = 3$).

[6] Eigenvalues with algebraic multiplicity 2 and 3 are not rare. For example a homography defined by pure shift $(\Delta x, \Delta y)$ has the form:

eigenvalue. Similarly, every linear combination of $\mathbf{u_{g_1}}$ and $\mathbf{u_{g_2}}$ is an eigenvector of $G$ with the same eigenvalue. Therefore:

$$H\mathbf{u_{b}}_j = \alpha_j \mathbf{u_{g}}_1 + \beta_j \mathbf{u_{g}}_2 \tag{14}$$

where $\alpha_j$ and $\beta_j$ are unknown scalars ($j = 1, 2$). Hence, each of the two eigenvectors $\mathbf{u_{b1}}$ and $\mathbf{u_{b2}}$ provides 3 linear equations and 2 new unknowns. Therefore, in total, together they provide 2 new linear constraints on $H$.
Case III: $dim(V) = 3$. In this case any vector is an eigenvector (all with the same eigenvalue $\lambda$). This is the case when $B \cong G \cong \lambda I$ are the identity transformation up to scale, i.e., no camera motion. In this case (as expected) $B$ and $G$ provide no additional constraints on $H$.

**Counting Constrains:**
So far we counted the number of constraints imposed on $H$ by a single eigen subspace. In order to count the total number of linear constraints that $B$ and $G$ impose on $H$, we analyze every possible combination of eigen subspaces according to the algebraic multiplicity of their eigenvalues:

1. $\lambda_i \neq \lambda_j \neq \lambda_k$. This implies $V_i \neq V_j \neq V_k$ and $dim(V_i) = dim(V_j) = dim(V_k) = 1$.

2. $\lambda_i = \lambda_j \neq \lambda_k$ ($V_i = V_j \neq V_k$). There are two such cases:
   (a) $dim(V_i = V_j) = 2$, and $dim(V_k) = 1$.
   (b) $dim(V_i = V_j) = 1$, and $dim(V_k) = 1$.

3. $\lambda_i = \lambda_j = \lambda_k$. In this case there is only a single eigen subspace $V = V_i = V_j = V_k$. Its dimensionality may be 1,2, or 3.

The following table summarizes the number of linearly independent constraints for each of the above cases:

| Case | **Eigenvalue Algebraic Multiplicity** | **Eigen Subspace Dimensionality** | **# of linearly independent constraints** |
|---|---|---|---|
| (1) | $\lambda_i \neq \lambda_j \neq \lambda_k$ | $\|V_i\| = \|V_j\| = \|V_k\| = 1$ | 6 |
| (2.a) | $\lambda_i = \lambda_j \neq \lambda_k$ | $\|V_i = V_j\| = 2, \|V_k\| = 1$ | 4 |
| (2.b) | $\lambda_i = \lambda_j \neq \lambda_k$ | $\|V_i = V_j\| = 1, \|V_k\| = 1$ | 4 |
| (3.a) | $\lambda_i = \lambda_j = \lambda_k$ | $\|V_i = V_j = V_k\| = 1$ | 2 |
| (3.b) | $\lambda_i = \lambda_j = \lambda_k$ | $\|V_i = V_j = V_k\| = 2$ | 2 |
| (3.c) | $\lambda_i = \lambda_j = \lambda_k$ | $\|V_i = V_j = V_k\| = 3$ | 0 |

To summarize: When $B$ (and $G$) have either two or three distinct eigenvalues (which is typical of general frame-to-frame transformations), then *two independent pairs of transformations suffice to uniquely determine* $H$. This is because each pair of transformations imposes 4 to 6 linearly independent constraints, and in theory 8 independent linear constraints suffice to uniquely resolve $H$ (up to arbitrary scale factor).

### 6.3 Numerical Stability

The final step in our algorithm is to solve a homogeneous set of linear equations (Eq. (8)). Care has to be taken when solving this system. For example, inaccuracies in the estimated frame-to-frame transformations decrease the

$H = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$. This matrix has a single eigenvalue $\lambda_1 = \lambda_2 = \lambda_3 = 1$ with algebraic multiplicity three. The corresponding eigen subspace has dimensionality 2. It is spanned by two linearly independent eigenvectors $[1, 0, 0]^t$ and $[0, 1, 0]^t$.

accuracy of the final output. The previous section showed that two independent pairs of transformations may suffice to uniquely determine $H$. In practice, however, to increase numerical stability, we use all available constraints from all pairs of *reliable* transformations after subsampling of the sequences, outlier rejection and normalization. These are explained next:

**Temporal Subsampling:** When the frame-to-frame transformations are too small, we often temporally subsample the sequences to obtain more significant transformations between successive frames. In our experiments where video clips were a couple of hundred frames long, we usually used 30 reliable transformations between distant (non-successive) frames. Such temporal subsampling should be done after recovering the temporal synchronization, to assure that it is done in a temporally synchronized manner across the two sequences.

**Outlier Rejection:** Inaccurate frame-to-frame transformations $T_i$ are pruned out by using two outlier detection mechanisms:
(i) The transformation between successive frames within each sequence are computed in both directions. Let $T_i$ be the transformation from $I_i$ to $I_{i+1}$, and $T_i^{Reverse}$ the transformation from $I_{i+1}$ to $I_i$. Then we measure the deviation of the composed matrix $T_i T_i^{Reverse}$ from the identity matrix in terms of the maximal induced residual misalignment of pixels, i.e.,

$$Reliability(T_i) = \max_{p \in I_i} ||T_i T_i^{Reverse} p - p|| \tag{15}$$

(ii) The similarity criterion of Eq. (5) can also be used to verify the degree of "similarity" between a pair of transformations $T_i$ and $T_i'$. After $\Delta t$ has been estimated and before $H$ is estimated, an unreliable pair of transformations can be detected and pruned out by measuring the deviation of $Sim(T_i, T_i')$ from 1. However, the first outlier criterion (that of Eq. (15)) proved to be more powerful.

**Matrix Normalization:** Using the heuristic provided in [11] (originly derived for Gaussian elimination) we normalize (scale) components of the input matrices $T_i$ and $T_i'$ in a way that the rows of the matrix $A$ of Eq. (8) will have approximately the same norm. This is an equivalent step to the scaling proposed by Hartley [13] for recovering fundamental matrices. This step indeed improve the results.

**Preferred Camera Motions:** When acquiring the sequences of input transformations, we usually have control over the camera motion. In general, any type of camera motion provides a frame-to-frame transformation which induces constraints on the inter-camera homography $H$. However, some transformations provide more stable sets of equations than others. In particular, we would like to generate sequences of transformations which provide more reliable components in each column of the matrix $A$ in Eq. (8). For example, image-plane rotations (i.e., rotations about the optical axis of one of the cameras) usually provide reliable entries in all columns of $M_i$ (a block of $A$), thus impose stable constraints on $H$. To conclude, the camera rig can (and should) be moved freely, however, it is recommended that a *few* of the camera movements include non-negligible image-plane rotations.

# 7 Conclusion

This paper presents an approach for aligning two sequences (both in time and in space), even when there is no common spatial information between the sequences. This was made possible by replacing the need for "consistent appearance" (which is a fundamental requirement in standard images alignment techniques), with the requirement of "consistent temporal behavior", which is often easier to satisfy. We demonstrated applications of this approach to real-world problems, which are inherently difficult for regular image alignment techniques.

## Acknowledgment

The authors would like to thank R. Basri and L. Zelnik-Manor for their helpful comments.

## References

[1] S. Avidan and A. Shashua. Thereading fundamaental matrices. In *European Conference on Computer Vision*, 1998.

[2] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequences. In *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695, 1996.

[3] A. Bjorck. *Numerical Methodes for Least Squares Problems*. SIAM, Philadelphia, 1996.

[4] P.R. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *International Conference on Computer Vision*, pages 173–182, 1993.

[5] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–689, Hilton Head Island, South Carolina, June 2000.

[6] Y. Caspi and M. Irani. Alignment of non-overlaping sequences. In *International Conference on Computer Vision*, volume II, Vancouver, Canada, 2001.

[7] D. Demirdijian, A. Zisserman, and R. Horaud. Stereo autocalibration from one plane. In *European Conference on Computer Vision*, 2000.

[8] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, June 2000.

[9] C. E.P̃earson (ed.). *Handbook of applied mathematics - Second Edition*. Van Nostrand Reinhold Company, New York, 1983.

[10] F. R. Gantmakher. *The theory of matrices*. Chelsea Pub., New York, 1959.

[11] Gene Golub and Charles Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, 1989.

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

[13] Richard I. Hartley. In defence of the 8-point algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[14] R. Horaud and G. Csurka. reconstruction using motions of a stereo rig. In *International Conference on Computer Vision*, pages 96–103, 1998.

[15] R. Horaud and F. Dornaika. Hand-eye calibration. *International Journal of Robotics Research*, 14(3):195–210, June 1995.

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

[17] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, pages 829–845, Freiburg, June 1998.

[18] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, January 1994.

[19] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: parallax based approach. In *International Conference on Pattern Recognition*, pages 685–688, 1994.

[20] Harpreet Sawhney. 3D geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[21] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Wa., June 1994.

[22] C.C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, 1980.

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

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

[25] R. Y. Tsai and R. K. Lenz. A new technique for full autonomous and efficient 3D robotics hand/eye calibration. *IEEE Journal of Robotics and Automation*, 5(3):345–358, June 1989.

[26] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *International Conference on Computer Vision*, pages 16–23, 1995.

[27] A. Zisserman, P.A. Beardsley, and I.D. Reid. Metric calibration of a stereo rig. In *Workshop on Representations of Visual Scenes*, pages 93–100, 1995.

# Increasing Space-Time Resolution in Video

Eli Shechtman, Yaron Caspi, Michal Irani

Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel
{elishe,caspi,irani}@wisdom.weizmann.ac.il

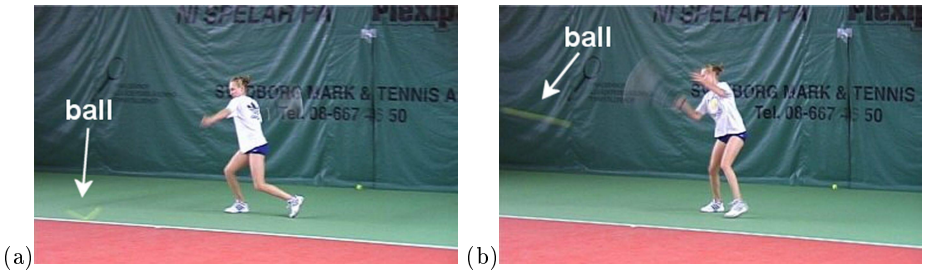
**Abstract.** We propose a method for constructing a video sequence of high space-time resolution by combining information from multiple low-resolution video sequences of the same dynamic scene. Super-resolution is performed simultaneously in time and in space. By "temporal super-resolution" we mean recovering rapid dynamic events that occur faster than regular frame-rate. Such dynamic events are not visible (or else observed incorrectly) in any of the input sequences, even if these are played in "slow-motion".
The spatial and temporal dimensions are very different in nature, yet are inter-related. This leads to interesting visual tradeoffs in time and space, and to new video applications. These include: (i) treatment of *spatial* artifacts (e.g., motion-blur) by increasing the *temporal* resolution, and (ii) combination of input sequences of different space-time resolutions (e.g., NTSC, PAL, and even high quality still images) to generate a high quality video sequence.

**Keywords:** Super-resolution (in time and space), Visual motion.

## 1 Introduction

A video camera has limited spatial and temporal resolution. The spatial resolution is determined by the spatial density of the detectors in the camera and by their induced blur. These factors limit the minimal size of spatial features or objects that can be visually detected in an image. The temporal resolution is determined by the frame-rate and by the exposure-time of the camera. These limit the maximal speed of dynamic events that can be observed in a video sequence.

Methods have been proposed for increasing the spatial resolution of images by combining information from multiple low-resolution images obtained at sub-pixel displacements (e.g. [1, 2, 5, 6, 8–12]. See [3] for a comprehensive review). These, however, usually assume static scenes and do not address the limited temporal resolution observed in dynamic scenes. In this paper we extend the notion of super-resolution to the *space-time* domain. We propose a unified framework for increasing the resolution both in time and in space by combining information from multiple *video sequences* of dynamic scenes obtained at (sub-pixel) spatial and (sub-frame) temporal misalignments. As will be shown, this enables new




(a) (b)

**Fig. 1. Motion blur.** *Distorted shape due to motion blur of very fast moving objects (the tennis ball and the racket) in a real tennis video. The perceived distortion of the ball is marked by a white arrow. Note, the "V"-like shape of the ball in (a), and the elongated shape of the ball in (b). The racket has almost "disappeared".*

visual capabilities of dynamic events, gives rise to visual tradeoffs between time and space, and leads to new video applications. These are substantial in the presence of very fast dynamic events.

Rapid dynamic events that occur faster than the frame-rate of video cameras are not visible (or else captured incorrectly) in the recorded video sequences. This problem is often evident in sports videos (e.g., tennis, baseball, hockey), where it is impossible to see the full motion or the behavior of the fast moving ball/puck. There are two typical visual effects in video sequences which are caused by very fast motion. One effect (motion blur) is caused by the exposure-time of the camera, and the other effect (motion aliasing) is due to the temporal sub-sampling introduced by the frame-rate of the camera:

(i) *Motion Blur:* The camera integrates the light coming from the scene during the exposure time in order to generate each frame. As a result, fast moving objects produce a noted blur along their trajectory, often resulting in distorted or unrecognizable object shapes. The faster the object moves, the stronger this effect is, especially if the trajectory of the moving object is not linear. This effect is notable in the distorted shapes of the tennis ball shown in Fig. 1. Note also that the tennis racket also "disappears" in Fig. 1.b. Methods for treating motion blur in the context of image-based super-resolution were proposed in [2, 11]. These methods however, require prior segmentation of moving objects and the estimation of their motions. Such motion analysis may be impossible in the presence of severe shape distortions of the type shown in Fig. 1. We will show that by increasing the *temporal resolution* using information from multiple video sequences, *spatial artifacts* such as motion blur can be handled without the need to separate static and dynamic scene components or estimate their motions.

(ii) *Motion-Based (Temporal) Aliasing:* A more severe problem in video sequences of fast dynamic events is false visual illusions caused by aliasing in time. Motion aliasing occurs when the trajectory generated by a fast moving object is characterized by frequencies which are higher than the frame-rate of the camera (i.e., the temporal sampling rate). When that happens, the high temporal frequencies are "folded" into the low temporal frequencies. The observable result is a distorted or even false trajectory of the moving object. This effect is illus-

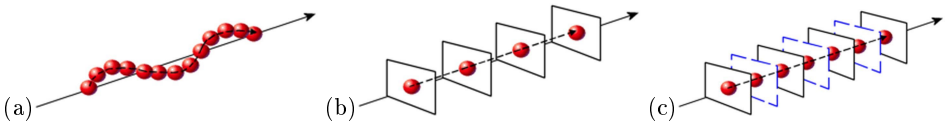



**Fig. 2. Motion aliasing.** *(a) shows a ball moving in a sinusoidal trajectory. (b) displays an image sequence of the ball captured at low frame-rate. The perceived motion is along a straight line. This false perception is referred to in the paper as "motion aliasing". (c) Illustrates that even using an ideal temporal interpolation will not produces the correct motion. The filled-in frames are indicated by dashed blue line.*

trated in Fig. 2, where a ball moves fast in sinusoidal trajectory of high frequency (Fig. 2.a). Because the frame-rate is much lower (below Nyquist frequency of the trajectory), the *observed* trajectory of the ball is a straight line (Fig. 2.b). Playing that video sequence in "slow-motion" will not correct this false visual effect (Fig. 2.c). Another example of motion-based aliasing is the well-known visual illusion called the "wagon wheel effect": When a wheel is spinning very fast, beyond a certain speed it will appear to be rotating in the "wrong" direction.

Neither the motion-based aliasing nor the motion blur can be treated by playing such video sequences in "slow-motion", even when sophisticated temporal interpolations are used to increase the frame-rate. This is because the information contained in a single video sequence is insufficient to recover the missing information of very fast dynamic events (due to excessive blur and subsampling). Multiple video sequences, on the other hand, provide additional samples of the dynamic space-time scene. While none of the individual sequences provides enough visual information, combining the information from all the sequences allows to generate a video sequence of high space-time resolution (Sec. 2), which displays the correct dynamic events. Thus, for example, a reconstructed high-resolution sequence will display the correct motion of the wagon wheel despite it appearing incorrectly in *all* of the input sequences (Sec. 3).

The spatial and temporal dimensions are very different in nature, yet are inter-related. This introduces visual tradeoffs between space and times, which are unique to spatio-temporal super-resolution, and are not applicable in traditional spatial (i.e., image-based) super-resolution. For example, output sequences of different space-time resolutions can be generated for the same input sequences. A large increase in the temporal resolution usually comes at the expense of a large increase in the spatial resolution, and vice versa.

Furthermore, input sequences of different space-time resolutions can be meaningfully combined in our framework. In traditional image-based super-resolution there is no incentive to combine input images of different spatial resolutions, since a high-resolution image will subsume the information contained in a low-resolution image. This, however, is not the case here. Different types of cameras of different space-time resolutions may provide *complementary* information. Thus, for example, we can combine information obtained by high-quality still cameras (which have very high spatial-resolution, but extremely low "temporal resolution"), with information obtained by standard video cameras (which have low

spatial-resolution but higher temporal resolution), to obtain an improved video sequence of high spatial and high temporal resolution. These issues and other space-time visual tradeoffs are discussed in Sec. 4.

## 2 Space-Time Super-Resolution

Let $S$ be a dynamic space-time scene. Let $\{S_i^l\}_{i=1}^n$ be $n$ video sequences of that dynamic scene recorded by $n$ different video cameras. The recorded sequences have limited spatial and temporal resolution. Their limited resolutions are due to the space-time imaging process, which can be thought of as a process of blurring followed by sampling in time and in space.

The blurring effect results of the fact that the color at each pixel in each frame (referred to as a "space-time point" and marked by the small pink or blue box in Fig. 3.a) is an integral (a weighted average) of the colors in a space-time *region* in the dynamic scene $S$ (marked by the large pink or blue boxes respectively in Fig. 3.a). The temporal extent of this region is determined by the exposure-time of the video camera, and the spatial extent of this region is determined by the spatial point-spread-function (PSF) of the camera (determined by the properties of the lens and the detectors [4]).

The sampling process also has a spatial and a temporal component. The spatial sampling results from the fact that the camera has a discrete and finite number of detectors (the output of each is a single pixel value), and the temporal sampling results from the fact that the camera has a finite frame-rate resulting in discrete frames (typically 25 $frames/sec$ in PAL cameras and 30 $frames/sec$ in NTSC cameras).

The above space-time imaging process inhibits high spatial and high temporal frequencies of the dynamic scene, resulting in video sequences of low space-time resolutions. Our objective is to use the information from all these sequences to construct a new sequence $S^h$ of high space-time resolution. Such a sequence will have smaller blurring effects and finer sampling in space and in time, and will thus capture higher space-time frequencies of the dynamic scene $S$. In particular, it will capture fine spatial features in the scene and rapid dynamic events which cannot be captured by the low-resolution sequences.

The recoverable high-resolution information in $S^h$ is limited by its spatial and temporal sampling rate (or discretization) of the space-time volume. These rates can be different in space and in time. Thus, for example, we can recover a sequence $S^h$ of very high spatial resolution but low temporal resolution (e.g., see Fig. 3.b), a sequence of very high temporal resolution but low spatial resolution (e.g., see Fig. 3.c), or a bit of both. These tradeoffs in space-time resolutions and their visual effects will be discussed in more detail later in Sec. 4.2.

We next model the geometrical relations (Sec. 2.1) and photometric relations (Sec. 2.2) between the unknown high-resolution sequence $S^h$ and the input low-resolution sequences $\{S_i^l\}_{i=1}^n$.

### 2.1 The Space-time Coordinate Transformations

In general a space-time dynamic scene is captured by a 4D representation $(x, y, z, t)$. For simplicity, in this paper we deal with dynamic scenes which can be modeled by a 3D space-time volume $(x, y, t)$ (see in Fig. 3.a). This assumption is valid if one of the following conditions holds: (i) the scene is planar and the dynamic events occur within this plane, or (ii) the scene is a general dynamic 3D scene, but the distances between the recording video cameras are small relative to their distance from the scene. (When the camera centers are very close to each other, there is no relative 3D parallax.) Under those conditions the dynamic scene can be modeled by a 3D space-time representation.

W.l.o.g., let $S_1^l$ be a "reference" sequence whose axes are aligned with those of the continuous space-time volume $S$ (the unknown dynamic scene we wish to reconstruct). $S^h$ is a discretization of $S$ with a higher sampling rate than that of $S_1^l$. Thus, we can model the transformation $T_1$ from the space-time coordinate system of $S_1^l$ to the space-time coordinate system of $S^h$ by a scaling transformation (the scaling can be different in time and in space). Let $T_{i\to 1}$ denote the space-time coordinate transformation from the reference sequence $S_1^l$ to the $i$-th low resolution sequence $S_i^l$ (see below). Then the space-time coordinate transformation of each low-resolution sequence $S_i^l$ is related to that of the high-resolution sequence $S^h$ by $T_i = T_1 \cdot T_{i\to 1}$.

The space-time coordinate transformation between two input sequences ($T_{i\to 1}$) results from the different setting of the different cameras. A *temporal misalignment* between two sequences occurs when there is a time-shift (offset) between them (e.g., if the cameras were not activated simultaneously), or when they differ in their frame rates (e.g., PAL and NTSC). Such temporal misalignments can be modeled by a 1-D affine transformation in time, and is typically at sub-frame time units. The *spatial misalignment* between the two sequences results from the fact that the two cameras have different external and internal calibration parameters. In our current implementation, as mentioned above, because the camera centers are assumed to be very close or else the scene is planar, the spatial transformation can thus be modeled by an inter-camera homography. We computed these space-time coordinate transformations, using the method of [7], which provides high sub-pixel and high sub-frame accuracy.

Note that while the space-time coordinate transformations *between the sequences* ($\{T_i\}_{i=1}^n$) are very simple (a spatial homography and a temporal affine transformation), the motions occurring over time *within* the dynamic scene can be very complex. Our space-time super-resolution algorithm does *not* require knowledge of these motions, only the knowledge of $\{T_i\}_{i=1}^n$. It can thus handle very complex dynamic scenes.

### 2.2 The Space-Time Imaging Model

As mentioned earlier, the space-time imaging process induces spatial and temporal blurring in the low-resolution sequences. The temporal blur in the low-

resolution sequence $S_i^l$ is caused by the exposer-time $\tau_i$ of the $i$-th camera. The spatial blur in $S_i^l$ is due to the spatial point-spread-function (PSF) of the $i$-th camera, which can be approximated by a 2D spatial Gaussian with std $\sigma_i$. (A method to estimate the PSF of a camera can be found in [10].)

Let $B_i = B_{(\sigma_i, \tau_i, p_i^l)}$ denote the combined space-time blur operator of the $i$-th camera corresponding to the low resolution space-time point $p_i^l = (x_i^l, y_i^l, t_i^l)$. Let $p^h = (x^h, y^h, t^h)$ be the corresponding high resolution space-time point $p^h = T_i(p_i^l)$ ($p^h$ is not necessarily an integer grid point of $S^h$, but is contained in the continuous space-time volume $S$). Then the relation between the *unknown* space-time values $S(p^h)$, and the *known* low resolution space-time measurements $S_i^l(p_i^l)$, can be expressed by:

$$S_i^l(p_i^l) = \left(S * B_i^h\right)(p^h) = \int_x \int_y \int_t \limits_{p\,=\,(x,\,y,\,t)\,\in\,Support(B_i^h)} S(p)\; B_i^h(p - p^h)dp \qquad (1)$$

where $B_i^h = T_i(B_{(\sigma_i, \tau_i, p_i^l)})$ is a point-dependent space-time blur kernel represented in the high resolution coordinate system. Its support is illustrated by the large pink or blue boxes in Fig. 3.a. To obtain a linear equation in the terms of the *discrete unknown* values of $S^h$ we used a discrete approximation of Eq. (1). In our implementation we used a non-isotropic approximation in the temporal dimension, and an isotropic approximation in the spatial dimension (see [6] for a discussion of the different discretization techniques in the context of image-based super-resolution ). Eq. (1) thus provides a linear equation that relates the unknown values in the high resolution sequence $S^h$ to the *known* low resolution measurements $S_i^l(p_i^l)$.

When video cameras of different photometric responses are used to produce the input sequences, then a preprocessing step that histogram-equalizes all the low resolution sequences is necessary. This step is required to guarantee consistency of the relation in Eq. (1) with respect to all low resolution sequences.

### 2.3 The Reconstruction Step

Eq. (1) provides a single equation in the high resolution unknowns for each low resolution space-time measurement. This leads to the following huge system of linear equations in the unknown high resolution elements of $S^h$:

$$A\,\vec{h} = \vec{l} \qquad (2)$$

where $\vec{h}$ is a vector containing all the unknown high resolution color values (in YIQ) of $S^h$, $\vec{l}$ is a vector containing all the space-time measurements from all the low resolution sequences, and the matrix $A$ contains the relative contributions of each high resolution space-time point to each low resolution space-time point, as defined by Eq. (1).

When the number of low resolution space-time measurements in $\vec{l}$ is greater than or equal to the number of space-time points in the high-resolution sequence

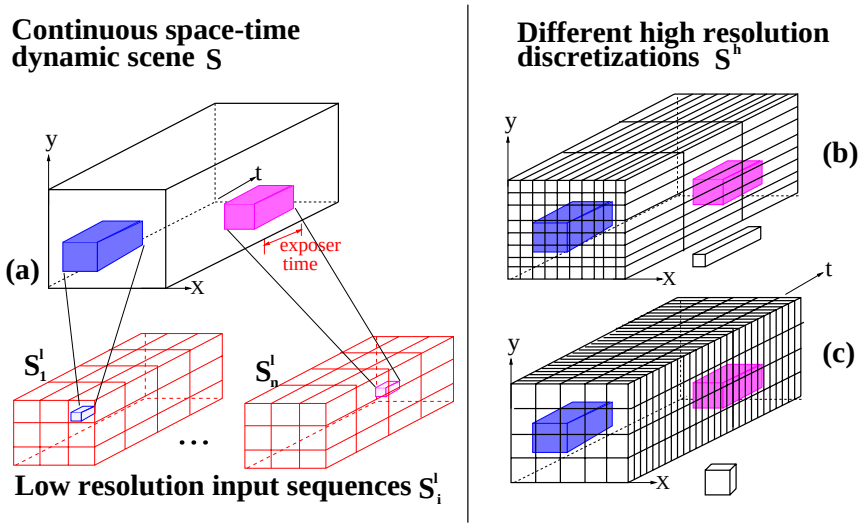



**Fig. 3. The space-time imaging process.** *(a) illustrates the space-time continuous scene and two of the low resolution sequences. The large pink and blue boxes are the support regions of the space-time blur corresponding to the low resolution space-time measurements marked by the small pink and blue boxes. (b,c) show two different possible discretizations of the space-time volume resulting in two different high resolution output sequences. (b) has a low frame-rate and high spatial resolution, (c) has a high frame-rate but low spatial resolution.*

$S^h$ (i.e., in $\overrightarrow{h}$), then there are more equations than unknowns, and Eq. (2) can be solved using LSQ methods. This, however, implies that a large increase in the spatial resolution (which requires very fine spatial sampling in $S^h$) will come at the expense of a significant increase in the temporal resolution (which also requires fine temporal sampling in $S^h$), and vice versa. This is because for a given set of input low-resolution sequences, the size of $\overrightarrow{l}$ is fixed, thus dictating the number of unknowns in $S^h$. It can, however, be distributed differently between space and time, resulting in different space-time resolutions (see 4.2).

**Directional space-time regularization:** When there is an insufficient number of cameras relative to the required improvement in resolution (either in the entire space-time volume, or only in portions of it), then the above set of equations (2) becomes ill-posed. To constrain the solution and provide additional numerical stability, a space-time regularization term can be added to impose smoothness on the solution $S^h$ in space-time regions which have insufficient information. We introduce a *directional* (or steerable [12]) space-time regularization term which applies smoothness only in directions where the derivatives are low, and does *not* smooth across space-time "edges". In other words, we seek $\overrightarrow{h}$ which minimize the following error term:

$$min\left(||A\overrightarrow{h} - \overrightarrow{l}||^2 + ||W_x L_x \overrightarrow{h}||^2 + ||W_y L_y \overrightarrow{h}||^2 + ||W_t L_t \overrightarrow{h}||^2\right) \qquad (3)$$

Where $L_j$ $(j = x, y, t)$ is matrix capturing the second derivative operator in the direction $j$, and $W_j$ is a diagonal weight matrix which captures the degree of desired regularization at each space-time point in the direction $j$. The weights in $W_j$ prevent smoothing across space-time "edges". These weights are determined by the location, orientation and magnitude of space-time edges, and are approximated using space-time derivatives in the low resolution sequences.

**Solving the equation:** The optimization problem of Eq. (3) has very large dimensionality. For example, even for a simple case of four low resolution input sequences, each one-second long (25 frames) and of size $128 \times 128$ pixels, we get: $128^2 \times 25 \times 4 \approx 1.6 \times 10^6$ equations from the low resolution measurements alone (without regularization). Assuming a similar number of high resolution unknowns poses a severe computational problem. However, matrix $A$ is sparse and local (i.e., all the non zero entries are located in a few diagonals), the system of equations can be solved using "block relaxation" [13].

## 3 Examples

**Empirical Evaluation:** To examine the capabilities of temporal super-resolution in the presence of strong motion aliasing and strong motion blur, we first simulated a sports-like scene with a very fast moving object. We recorded a single video sequence of a basketball bouncing on the ground. To simulate high speed of the ball relative to frame-rate and relative to the exposure-time (similar to those shown in Fig. 1), we temporally blurred the sequence using a large (9-frame) blur kernel, followed by a large subsampling in time by factor of 30. This process results in a low temporal-resolution sequences of a very fast dynamic event having an "exposure-time" of about $\frac{1}{3}$ of its frame-time. We generated 18 such low resolution sequences by starting the temporal sub-sampling at arbitrary starting frames. Thus, the input low-resolution sequences are related by *non-uniform* sub-frame temporal offsets. Because the original sequence contained 210 frames, each generated low-resolution sequence contains only 7 frames. Three of the 18 sequences are presented in Fig 4.a-c. To visually display the event captured in each of these sequences, we super-imposed all 7 frames in each sequence. Each ball in the super-imposed image represents the location of the ball at a different frame. None of the 18 low resolution sequences captures the correct trajectory of the ball. Due to the severe motion aliasing, the perceived ball trajectory is roughly a smooth curve, while the true trajectory was more like a cycloid (the ball jumped 5 times on the floor). Furthermore, the shape of the ball is completely distorted in all input image frames, due to the strong motion blur.

We applied the super-resolution algorithm of Sec. 2 on these 18 low-resolution input sequences, and constructed a high-resolution sequence whose frame-rate is 30 times higher than that of the input sequences. (In this case we requested an increase only in the temporal sampling rate). The reconstructed high-resolution sequence is shown in Fig. 4.d. This is a super-imposed display of some of the reconstructed frames (every 8'th frame). The true trajectory of the bouncing ball has been recovered. Furthermore, Figs. 4(e)-(f) show that this process has

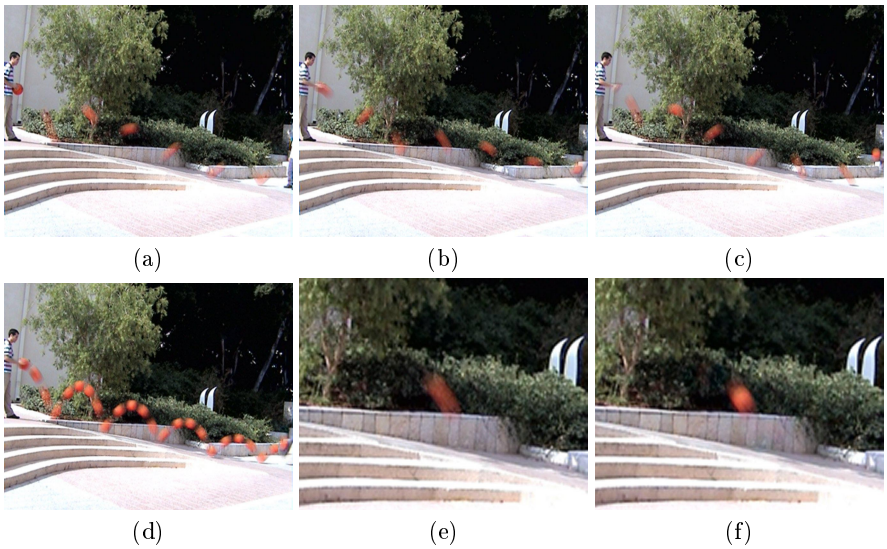



**Fig. 4. Temporal super-resolution.** *We simulated 18 low-resolution video recordings of a rapidly bouncing ball inducing strong motion blur and motion aliasing (see text). (a)-(c) Display the dynamic event captured by three representative low-resolution sequences. These displays were produced by super-position of all 7 frames in each low-resolution sequences. All 18 input sequences contain severe motion aliasing (evident from the falsely perceived curved trajectory of the ball) and strong motion blur (evident from the distorted shapes of the ball). (d) The reconstructed dynamic event as captured by the generated high-resolution sequence. The true trajectory of the ball is recovered, as well as its correct shape. (e) A close-up image of the distorted ball in one of the low resolution frames. (f) A close-up image of the ball at the exact corresponding frame in time in the high-resolution output sequence.*

removed almost all effects of motion blur and the true shape of moving ball has been automatically recovered, although no single low resolution frame contains the true shape of the ball. Note that no estimation of the ball motion was needed to obtain these results. This effect is explained in more details in Sec. 4.1.

The above results obtained by temporal super-resolution cannot be obtained by playing any low-resolution sequence in "slow-motion" due to the strong motion aliasing. Such results cannot be obtained either by interleaving frames from the 18 input sequences, due to the non-uniform time shifts between the sequences and due to the severe motion-blur observed in the individual image frames.

**A Real Example - The "wagon-wheel effect":** We used four independent PAL video cameras to record a scene of a fan rotating clock-wise very fast. The fan rotated faster and faster, until at some stage it exceeded the maximal velocity that can be captured by video frame-rate. As expected, at that moment all four input sequences display the classical "wagon wheel effect" where the fan appears to be falsely rotating backwards (counter clock-wise). We computed the spatial and temporal misalignments between the sequences at sub-pixel and sub-frame accuracy using [7] (the recovered temporal misalignments are displayed

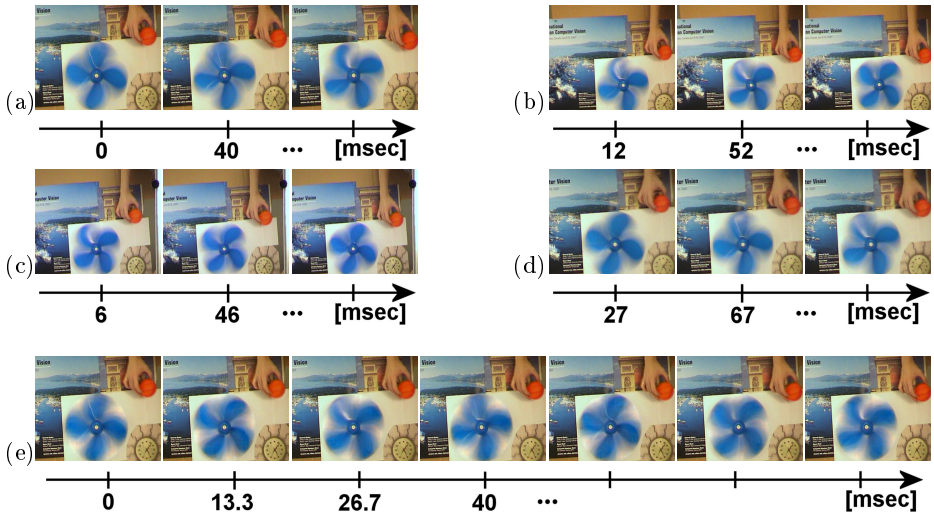



**Fig. 5. Temporal super-resolution (the "wagon wheel effect").** *(a)-(d) display 3 successive frames from four PAL video recordings of a fan rotating clock-wise. Because the fan is rotating very fast (almost* $90^o$ *between successive frames), the motion aliasing generates a false perception of the fan rotating slowly in the opposite direction (counter clock-wise) in all four input sequences. The temporal misalignments between the input sequences were computed at sub-frame temporal accuracy, and are indicated by their time bars. The spatial misalignments between the sequences (e.g., due to differences in zoom and orientation) were modeled by a homography, and computed at sub-pixel accuracy. (e) shows the reconstructed video sequence in which the temporal resolution was increased by a factor of 3. The new frame rate (*$75\frac{frames}{sec}$*) is also indicated by a time bars. The correct clock-wise motion of the fan is recovered.* **Please view attached video clips to perceive the strong dynamic effects.**

in Fig. 5.a-d using a time-bar). We used the super-resolution method of Sec. 2 to increase the temporal resolution by a factor of 3 while maintaining the same spatial resolution. The resulting high-resolution sequence displays the true forward (clock-wise) motion of the fan. Example of a few successive frames from each low resolution input sequence are shown in Fig.5.a-d for the portion where the fan appears to be rotating counter clock-wise. A few successive frames from the reconstructed high temporal-resolution sequence corresponding to the same time are shown in Fig.5.e, showing the correctly recovered (clock-wise) motion. It is difficult to perceive these strong dynamic effects via a static figure (Fig. 5). We therefore urge the reviewers to view the attached video clips where these effects are very vivid . Furthermore, playing the input sequences in "slow-motion" (using any type of temporal interpolation) will *not* reduce the perceived false motion effects. This is also shown in the attached video clips.

## 4 Space-Time Visual Tradeoffs

The spatial and temporal dimensions are very different in nature, yet are interrelated. This introduces visual tradeoffs between space and time, which are unique to spatio-temporal super-resolution, and are not applicable to traditional spatial (i.e., image-based) super-resolution.

### 4.1 Temporal Treatment of Spatial Artifacts

When an object moves fast relative to the exposure time of the camera, it induces observable motion-blur (e.g., see Fig. 1). The perceived distortion is spatial, however the cause is temporal. We next show that by increasing the *temporal* resolution we can handle the *spatial* artifacts caused by motion blur.

Motion blur is caused by the extended temporal blur due to the exposure-time. To decrease effects of motion blur we need to decrease the temporal blur, i.e., recover high temporal frequencies. This requires increasing the frame-rate beyond that of the low resolution input sequences. In fact, to decrease the effect of motion blur, the output temporal sampling rate must be increased so that the distance between the new high resolution temporal samples is *smaller* than the original exposure time of the low resolution input sequences.

This indeed was the case in the experiment of Fig. 4. Since the simulated exposure time in the low resolution sequences was $1/3$ of frame-time, an increase in temporal sampling rate by a factor $> 3$ effectively reduces the motion blur. The larger the increase the more effective the motion deblurring would be. This increase is limited, of course, by the number of input cameras.

A method for treating motion blur in the context of *image-based* super-resolution was proposed by [2, 11]. However, these methods require a prior segmentation of moving objects and the estimation of their motions. These methods will have difficulties handling complex motions or motion aliasing. The distorted shape of the object due to strong blur (e.g., Fig. 1) will pose severe problems in motion estimation. Furthermore, in the presence of motion aliasing, the direction of the estimated motion will not align with the direction of the induced blur. For example, the motion blur in Fig. 4.a-c. is along the true trajectory and not along the perceived one. In contrast, our approach does not require separation of static and dynamic scene components, nor their motion estimation, thus can handle very complex scene dynamics. However, we require multiple cameras.

Temporal frequencies in video sequences have very different characteristics than spatial frequencies, due to the different characteristics of the temporal and the spatial blur. The typical support of the spatial blur (PSF) is of a few pixels ($\sigma > 1$ *pixel*), whereas the exposure time is usually smaller than a single frame-time ($\tau <$ frame-time). Therefore, if we do not increase the output temporal sampling-rate *enough*, we will not improve the temporal resolution. In fact, if we increase the temporal sampling-rate a little but not beyond $\frac{1}{exposure\ time}$ of the low resolution sequences, we may even introduce *additional* motion blur. This dictates the number of input cameras needed for an effective decrease in the

motion-blur. An example of a case where an insufficient increase in the temporal sampling-rate introduced additional motion-blur is shown in Fig. 6.c3.

### 4.2 Producing Different Space-Time Outputs

In standard spatial super-resolution the increase in sampling rate is equal in all spatial dimensions. This is necessary in order to maintain the aspect ratio of image pixels, and to prevent distorted-looking images. However, this is not the case in space-time super-resolution. As explained in Sec. 2, the increase in sampling rate in the spatial and temporal dimensions need not be the same. Moreover, increasing the sampling rate in the spatial dimension comes at the expense of increase in the temporal frame rate, and vice-versa. This is because the number of unknowns in the high-resolution space-time volume depends on the space-time sampling rate, whereas the number of equations provided by the low resolution measurements remains fixed.

For example, assume that 8 video cameras are used to record a dynamic scene. One can increase the spatial sampling rate alone by a factor of $\sqrt{8}$ in x and y, or increase the temporal frame-rate alone by a factor of 8, or do a bit of both: increase the sampling rate by a factor of 2 in all three dimensions. Such an example is shown in Fig. 6. Fig. 6.a1 displays one of 8 low resolution input sequences. (Here we used only 4 video cameras, but split them into 8 sequences of even and odd fields). Figs. 6.a2 and 6.a3 display two possible outputs. In Fig. 6.a2 the increase is by a factor of 8 in the temporal axis with no increase in the spatial axes, and in Fig. 6.a3 the increase is by a factor of 2 in all axes x,y,t. Rows (b) and (c) illustrate the corresponding visual tradeoffs. The "$\times 1 \times 1 \times 8$" option (column 2) decreases the motion blur of the moving object (the toothpaste in (c.2)), while the "$\times 2 \times 2 \times 2$" option (column 3) improves the spatial resolution of the static background (b.3), but increases the motion blur of the moving object (c.3). The latter is because the increase in frame rate was only by factor 2 and did not exceed $\frac{1}{exposure\ time}$ of the video camera (see Sec. 4.1). In order to create a significant improvement in all dimensions, more than 4 video cameras are needed.

### 4.3 Combining Different Space-Time Inputs

So far we assumed that all input sequences were of similar spatial and temporal resolutions. The space-time super-resolution algorithm of Sec. 2 is not restricted to this case, and can handle input sequences of varying space-time resolutions. Such a case is meaningless in image-based super-resolution, because a high resolution input image would always contain the information of a low resolution image. In space-time super-resolution however, this is not the case. One camera may have high spatial but low temporal resolution, and the other vice-versa. Thus, for example, it is meaningful to combine information from NTSC and PAL video cameras. NTSC has higher temporal resolution than PAL ($30f/sec$ vs. $25f/sec$), but lower spatial resolution (640×480 pixels vs. 768×576 pixels). An

extreme case of this idea is to combine information from *still* and *video* cameras. Such an example is shown in Fig. 7. Two high quality still images (Fig. 7.a) of high spatial resolutions ($1152 \times 864$ pixels) but extremely low "temporal resolution" (the time gap between the two still images was 1.4 sec), were combined with an interlaced (PAL) video sequence (Fig. 7.b) using the algorithm of Sec 2. The video sequence has 3 times lower spatial resolution (we used fields of size $384 \times 288$ pixels), but a high temporal resolution ($50 f/sec$). The goal is to construct a new sequence of high spatial and high temporal resolutions ($1152 \times 864$ pixels at 50 $images/sec$). The output sequence shown in Fig. 7.c contains the high spatial resolution from the still images (the sharp text) and the high temporal resolution from the video sequence (the rotation of the toy dog and the brightening and dimming of illumination).

In the example of Fig. 7 we used only one input sequence and two still images, thus did not exceed the temporal resolution of the video or the spatial resolution of the stills. However, when multiple video cameras and multiple still images are used, the number of input measurements will exceed the number of output high resolution unknowns. In such cases the output sequence will exceed the spatial resolution of the still images and temporal resolution of the video sequences.

In Fig. 7 the number of unknowns was significantly larger than the number of low resolution measurements (the input video and the two still images). Yet, the reconstructed output was of high quality. The reason for this is the following: In video sequences the data is significantly more redundant than in images, due to the additional time axis. This redundancy provides more flexibility in applying *physically meaningful* directional regularization. In regions that have high spatial resolution but small (or no) motion (such as in the sharp text in Fig. 7), strong *temporal* regularization can be applied without decreasing the space-time resolution. Similarly, in regions with dynamic changes but low spatial resolution (such as in the rotating toy in Fig. 7), strong *spatial* regularization can be employed without degradation in space-time resolution. More generally, because a video sequence has much more data redundancy than an image has, the use of *directional space-time regularization* in video-based super-resolution is physically more meaningful and gives rise to recovery of higher space-time resolution than that obtainable by image-based super-resolution with image-based regularization.

## References

1. S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *CVPR*, Hilton Head Island, South Carolina, June 2000.
2. B. Bascle, A. Blake, and A.Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, pages 312–320, 1996.
3. S. Borman and R. Stevenson. Spatial resolution enhancement of low-resolution image sequences - a comprehensive review with directions for future research. Technical report, Laboratory for Image and Signal Analysis (LISA), University of Notre Dame, Notre Dame, July 1998.
4. M. Born and E. Wolf. *Principles of Optics*. Permagon Press, 1965.

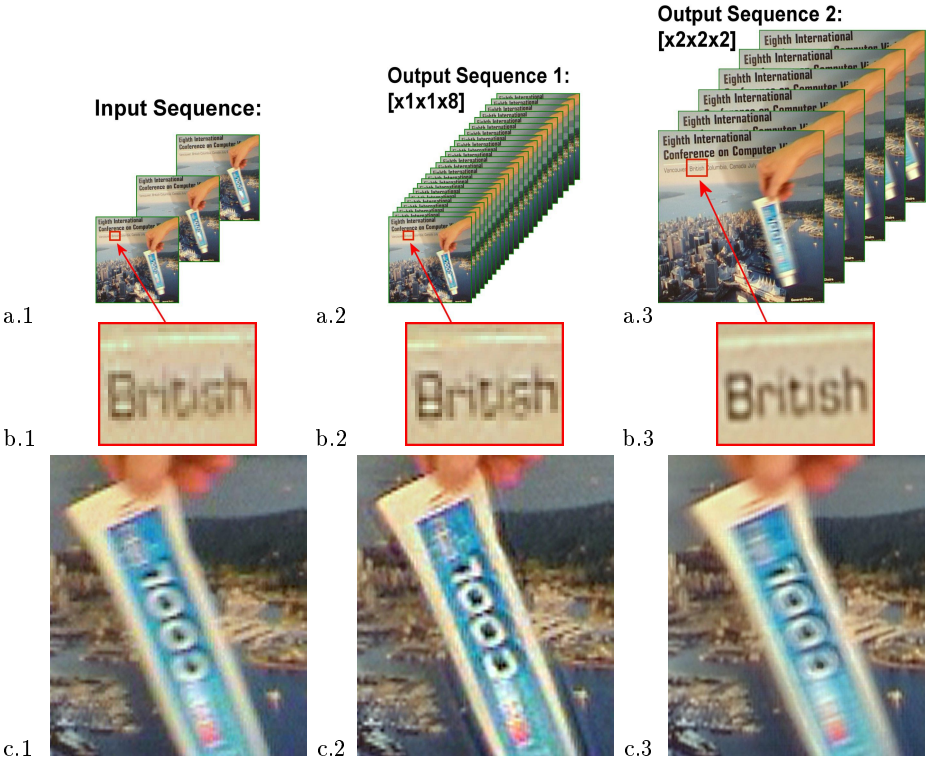



**Fig. 6. Tradeoffs between spatial and temporal resolution.** *This figure compares the visual tradeoffs resulting from applying space-time super-resolution with different discretization of the space-time volume. (a.1) displays one of eight low-resolution input sequences of a toothpaste in motion against a static background. (b.1) shows a close-up image of a static portion of the scene (the writing on the poster), and (c.1) shows a dynamic portion of the scene (the toothpaste). Column 2 (a.2, b.2, c.2) displays the resulting spatial and temporal effects of applying super-resolution by a factor of 8 in time only. Motion blur of the toothpaste is decreased. Column 3 (a.3, b.3, c.3) displays the resulting spatial and temporal effects of applying super-resolution by a factor of 2 in all three dimensions $x, y, t$. The spatial resolution of the static portions is increased (see "British" and the yellow line above it in b.3), but the motion blur is also increased (c.3). See text for an explanation of these visual tradeoffs.*

5. D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *CVPR*, pages 885–891, June 1998.
6. D. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *ICPR*, pages 600–605, 2000.
7. Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *CVPR*, pages 682–689, Hilton Head Island, South Carolina, June 2000.
8. M. Elad. Super-resolution reconstruction of images. Ph.D. Thesis, Technion Israel Institute of Technology, December 1996.
9. T.S. Huang and R.Y. Tsai. Multi-frame image restoration and registration. In *Advances in Computer Vision and Image Processing*, volume 1, pages 317–339.

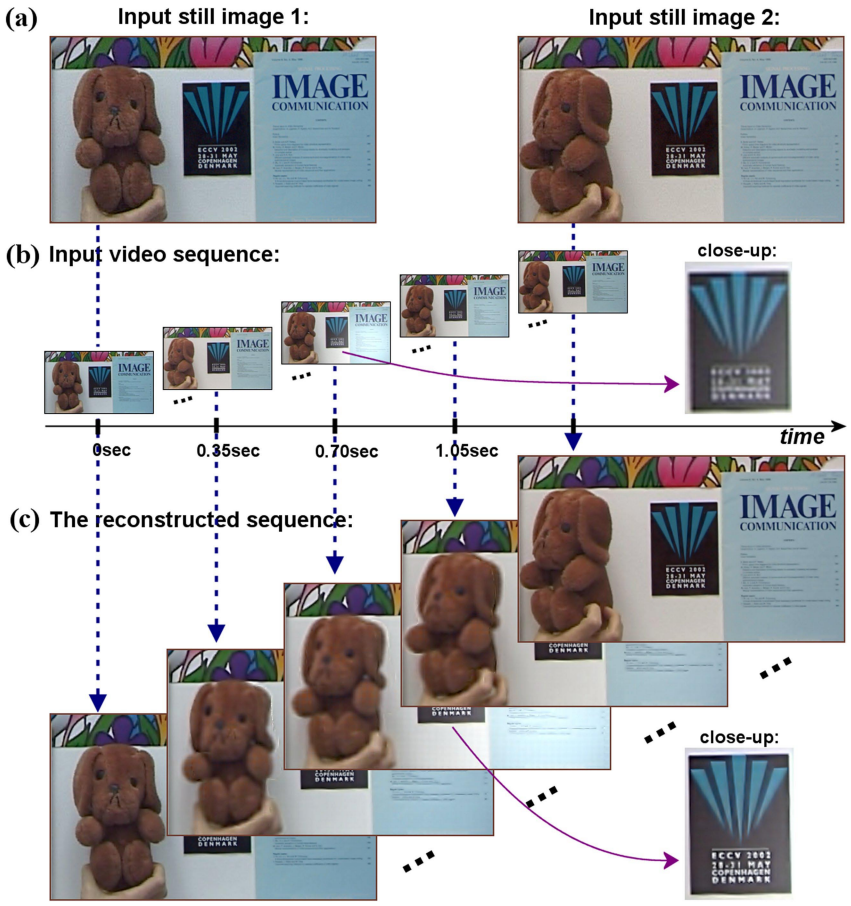



**Fig. 7. Combining still and video.** *A dynamic scene of a rotating toy-dog and varying illumination was captured by: (a) A still camera with spatial resolution of* $1152\times864$ *pixels, and (b) A video camera with* $384\times288$ *pixels at 50 f/sec. The video sequence was* $1.4$*sec long (*$70$ *frames), and the still images were taken* $1.4$*sec apart (together with the first and last frames). The algorithm of Sec. 2 is used to generate the high resolution sequence (c). The output sequence has the spatial dimensions of the still images and the frame-rate of the video (*$1152\times864\times50$*). It captures the temporal changes correctly (the rotating toy and the varying illumination), as well the high spatial resolution of the still images (the sharp text, see close-ups). Due to lack of space we show only a portion of the images, but the proportions between video and still are maintained.*

JAI Press Inc., 1984.
10. M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP:GM*, 53:231–239, May 1991.
11. A. J. Patti, M. I. Sezan, and A. M. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. In *IEEE Trans. on Image Processing*, volume 6, pages 1064–1076, August 1997.
12. J. Shin, J. Paik, J. R. Price, and M.A. Abidi. Adaptive regularized image interpolation using data fusion and steerable constraints. In *SPIE Visual Communications*

*and Image Processing*, volume 4310, January 2001.
13. U. Trottenber, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2000.

# Feature-Based Sequence-to-Sequence Matching*

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

For a more detailed version of this paper see http://www.wisdom.weizmann.ac.il/∼vision/traj2traj.html

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

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

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

The advantage of 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 [Stein 1998] and Lee et.al. [Lee et al. 2000] described a method for estimating a time shift and a homography between two sequences based on alignment of centroids of moving objects. However, in [Stein 1998; Lee et al. 2000] the centroids were treated as an *unordered* collection of feature points and not as trajectories. In contrast, we enforce correspondences between *trajectories*, thus avoiding the combinatorial complexity of establishing point matches of all points in all frames, resolving ambiguities in point correspondences, and allowing for temporal correspondences at *sub-frame* accuracy. This is not possible when the points are treated independently (i.e., as a "cloud of points").

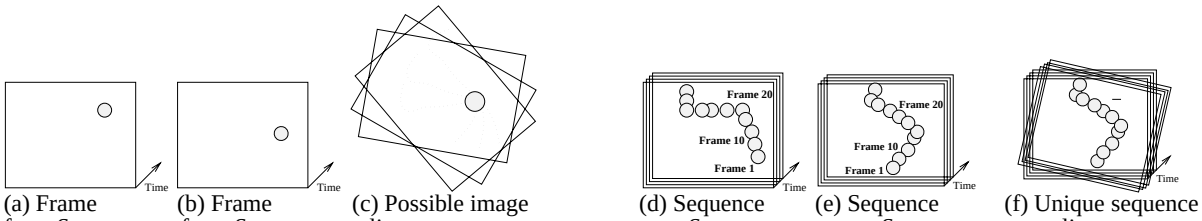
Our algorithm for recovering correspondences between trajectories across the two sequences is briefly described next. However, the ideas presented in this paper are not limited to this particular implementation.

**Implementation:**
Our current implementation is an extension of standard feature-based image matching methods (see examples of RANSAC/LMS-based methods in [Hartley and Zisserman 2000; Xu and Zhang 1996]). The first (and crucial) difference is that we use *trajectories* instead of *points* as our features. Since one trajectory consists of many points, therefore a single trajectory match induces multiple point matches (consequently, reducing the complexity of matching and increasing robustness in presence of errors – see "Benefits of the Approach").

A matching pair of 2D trajectory-features should correspond to projections of the same 3D trajectory of some 3D point. This 3D point need not be visible in the images (it can be real or virtual). For example, in our experiments we tracked moving objects (using background subtraction method) and extracted specific points on their blobs (e.g., the object centroid, or the highest point on the object silhouette, etc). The accuracy of approximating (real or virtual) 3D points from such 2D points on silhouettes is discussed in [Lee et al. 2000] and [Wong and Cipolla 2001].

The second difference (from standard feature based image matching implementations) is that we also deal with the temporal dimension to recover temporal matching as well. Schematically, the algorithm operates as follows: it searches in the space of possible trajectory correspondences (by a robust method, such as RANSAC or LMS). Each candidate trajectory correspondence is used for estimating spatial (homography $H$ or fundamental matrix $F$) and temporal ($\Delta t$) parameters by iterating the following two steps:
(i) Fix $\Delta t$ and approximate $H$ (or $F$) using standard methods.
(ii) Fix $H$ (or $F$) and refine $\Delta t$ by fitting the best linear interpolation

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




(a) Frame from $S_1$ (b) Frame from $S_2$ (c) Possible image alignments (d) Sequence $S_1$ (e) Sequence $S_2$ (f) Unique sequence alignment

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


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

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

value (we allow for sub-frame time shifts).
We then choose the spatial ($H$ or $F$) and temporal ($\Delta t$) candidate parameters which minimize the overall error. For more details see the long paper.

**Benefits of the Approach:**
(i) Trajectory matching requires only a single correct "feature" (i.e., trajectory) correspondence, as opposed to 8 feature (point) correspondences as in regular image-to-image matching (for estimating the fundamental matrix). This provides a significant benefit in RANSAC-like matching algorithms when the probability to select at random a sample of eight correct point correspondences is low. Such cases occur in wide-baseline scenarios where the range of valid disparities is very large. A complete analysis of the complexity reduction due to the smaller number of required "feature" (trajectory) matches may be found in the longer version of this paper.
(ii) Since trajectory-features can be constructed from "virtual 3D points" our method can address cases where the cameras never image the same scene points (e.g., when the cameras are on opposite sides of the scene, such as in Fig. 5).
(iii) Often corresponding feature points do not have similar appearance properties across cameras such as in the case of multi-sensor modalities (e.g., Fig 3), or in significantly different zooms (Fig. 4). Yet, their trajectories share common geometric/shape properties that facilitate the matching (e.g., see Fig. 2) even when the appearance properties are different .
(iv) Unsynchronized video sequences can be temporally matched (synchronized) at *sub-frame* accuracy. Such sub-frame synchronization gives rise to new video applications including super-resolution in time [Shechtman et al. 2002].
(v) Sub-frame temporal alignment also provides higher accuracy in the spatial matching. Image-to-image matching is restricted to matching of existing physical image frames. However, when "corresponding" frames in time across the two sequences have not been recorded at exactly the same time (due to a *sub-frame* temporal misalignment between the two sequences), this leads to inaccuracies in the spatial matching (fundamental matrix or homography). Sequence-to-sequence matching, on the other hand, is not restricted to physical ("integer") image frames.

**Examples:**
(i) Multi-sensor alignment: Fig. 3 shows results of aligning sequences obtained by two cameras of different sensing modalities. Fig. 3.(a) and 3.(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 3.(c). 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 is displayed in 3.(d). The recovered temporal misalignment was 1.31 sec. The results are displayed after fusing the two input sequences (using Burt's fusion algorithm [Burt and Kolczynski 1993]). We can now clearly observe spatial features from both sequences. In particular note the right kite which is more clearly visible in the visible-light sequence, and the left kite which is more clearly visible in the IR sequence (both marked by circles).
(ii) Matching across significant zoom differences: Fig. 4 shows an example of aligning sequences obtained at significantly different zooms. Fig. 4.(a) and 4.(b) display two representative frames from the reference sequence and second sequence, showing a ball thrown from side to side. The sequence in column 4.(a) was captured by a wide field-of-view camera, while the sequence in column 4.(b) was captured by a narrow field-of-view camera. The cameras where located next to each other (the spatial transformation was modeled by a homography) and the ratio in zooms was approximately $1:3$. The two sequences capture features at significantly different spatial resolutions, which makes the problem of inter-camera image-to-image alignment very difficult. The dynamic information (the trajectory of the ball's center of gravity), on the other hand, forms a powerful cue for alignment both in time and in space. Column 4.(c) displays superposition of corresponding frames *after* spatio-temporal alignment. The dark pink boundaries in

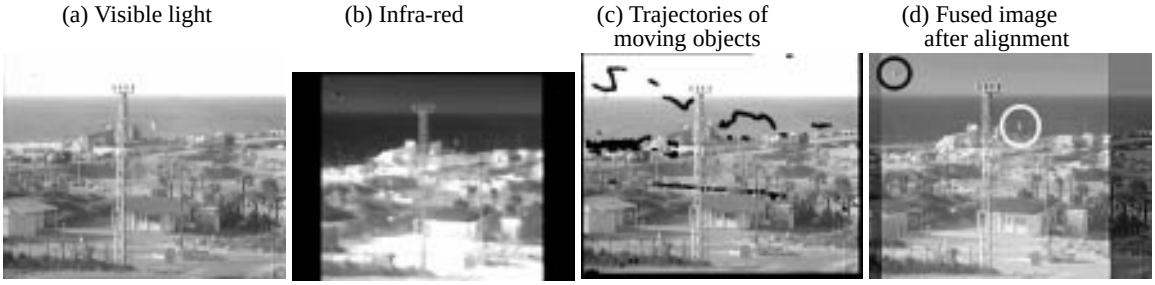



Figure 3: **Multi-Sensor Alignment** (see text).

4.(c) correspond to scene regions observed only by the reference (zoomed-out) camera.

(iii) Wide base-line matching: Fig. 5 shows an example of recovering the fundamental matrix using two cameras situated on opposite sides of the scene (i.e., the cameras are facing each other). Figs 5.(a) and 5.(b) display two representative frames from two sequences.Each camera is visible by the other camera and is circled and marked by a white arrow. Space-time trajectories induced by moving objects (ball and players) are displayed in 5.(c)-(d) in different colors for the different objects. The feature points that correspond to the current frame are marked in yellow. The recovered epipolar geometry is displayed in 5.(e) and 5.(f). Points and their epipolar lines are displayed in each image for verification. Note, that the only static objects that are visible in both views are the basket ring and the board. Accuracy of the recovered spatial alignment can be appreciated by the closeness of each point to the epipolar line of its corresponding point, as well as by comparing the intersection of epipolar lines with the ground truth epipole marked by a cross (which is the other camera). In this example the relative blob size of the moving objects was used to provide initial correspondence between the trajectories across the two sequences. Two trajectories (instead of one) were used on each RANSAC iteration, as most trajectories are planar. An initial temporal alignment with accuracy within one second (25 frames) was manually provided, and the final recovered temporal shift was 3.69 frames.

**Summary:**
We have shown that similar to [Caspi and Irani 2000] (where direct *intensity-based* image alignment was extended to *sequence alignment*), feature-based image matching can also be extended into trajectory-based sequence matching. This allows to address scenarios that are very difficult to solve otherwise.

**For a more detailed version and example sequences see** `www.wisdom.weizmann.ac.il/~vision/traj2traj.html`

## References


BERGEN, J., ANANDAN, P., HANNA, K., AND HINGORANI, R. 1992. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV)*, 237–252.

BURT, P., AND KOLCZYNSKI, R. 1993. Enhanced image capture through fusion. In *International Conference on Computer Vision (ICCV)*, 173–182.

CASPI, Y., AND IRANI, M. 2000. A step towards sequence-to-sequence alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 682–689.

FAUGERAS, O., LUONG, Q., AND PAPADOPOULO, T. 2001. *The Geometry of Multiple Images*. MIT Press.

HARTLEY, R., AND ZISSERMAN, A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge university press, Cambridge.

LEE, L., ROMANO, R., AND STEIN, G. 2000. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 22*, Special Issue on Video Surveillance and Monitoring (August), 758–767.

SHECHTMAN, E., CASPI, Y., AND IRANI, M. 2002. Increasing video resolution in time and space. In *European Conference on Computer Vision (ECCV)*.

STEIN, G. P. 1998. Tracking from multiple view points: Self-calibration of space and time. In *DARPA IU Workshop*, 1037–1042.

SZELISKI, R., AND SHUM, H.-Y. 1997. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics Proceedings, Annual Conference Series*, 251–258.

WONG, K.-Y. K., AND CIPOLLA, R. 2001. Structure and motion from silhouettes. In *International Conference on Computer Vision (ICCV)*, vol. II, 217–222.

XU, C., AND ZHANG, Z. 1996. *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers, Dordecht, The Netherlands.

ZHANG, Z., DERICHE, R., FAUGERAS, O., AND LUONG, Q. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence 78*, 87–119.

ZOGHLAMI, I., FAUGERAS, O., AND DERICHE, R. 1997. Using geometric corners to build a 2d mosaic from a set of images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 420–425.

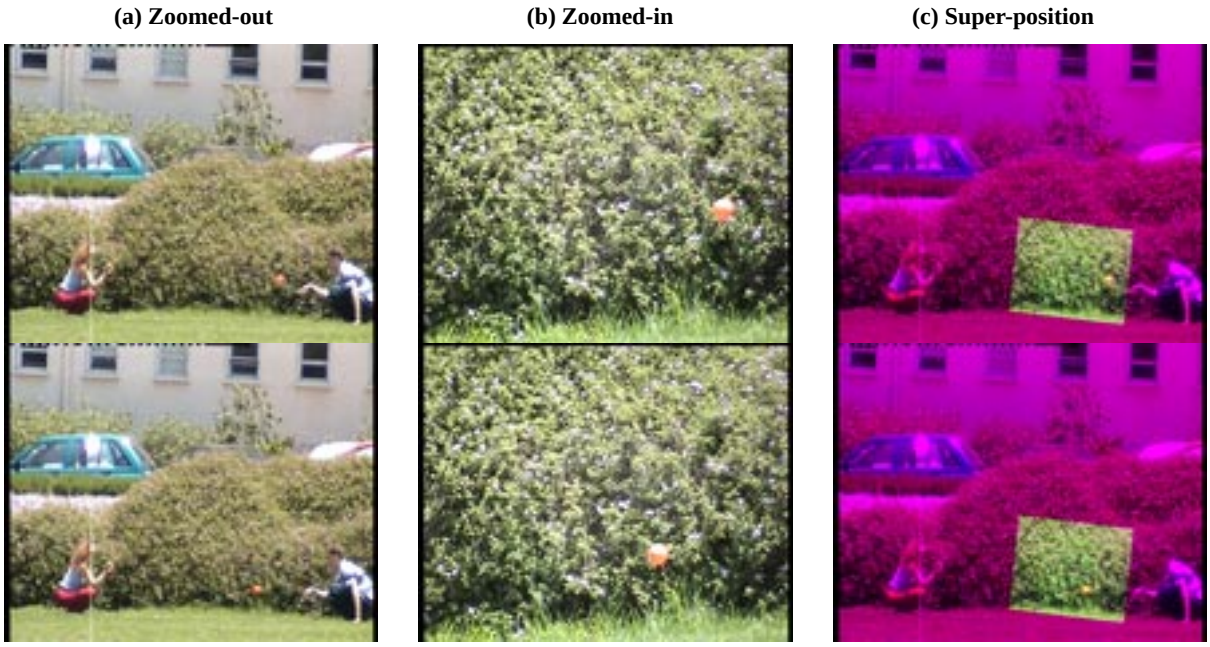



Figure 4: **Alignment of sequences obtained at different zooms** (see text).
For color sequences see `www.wisdom.weizmann.ac.il/~vision/traj2traj.html`

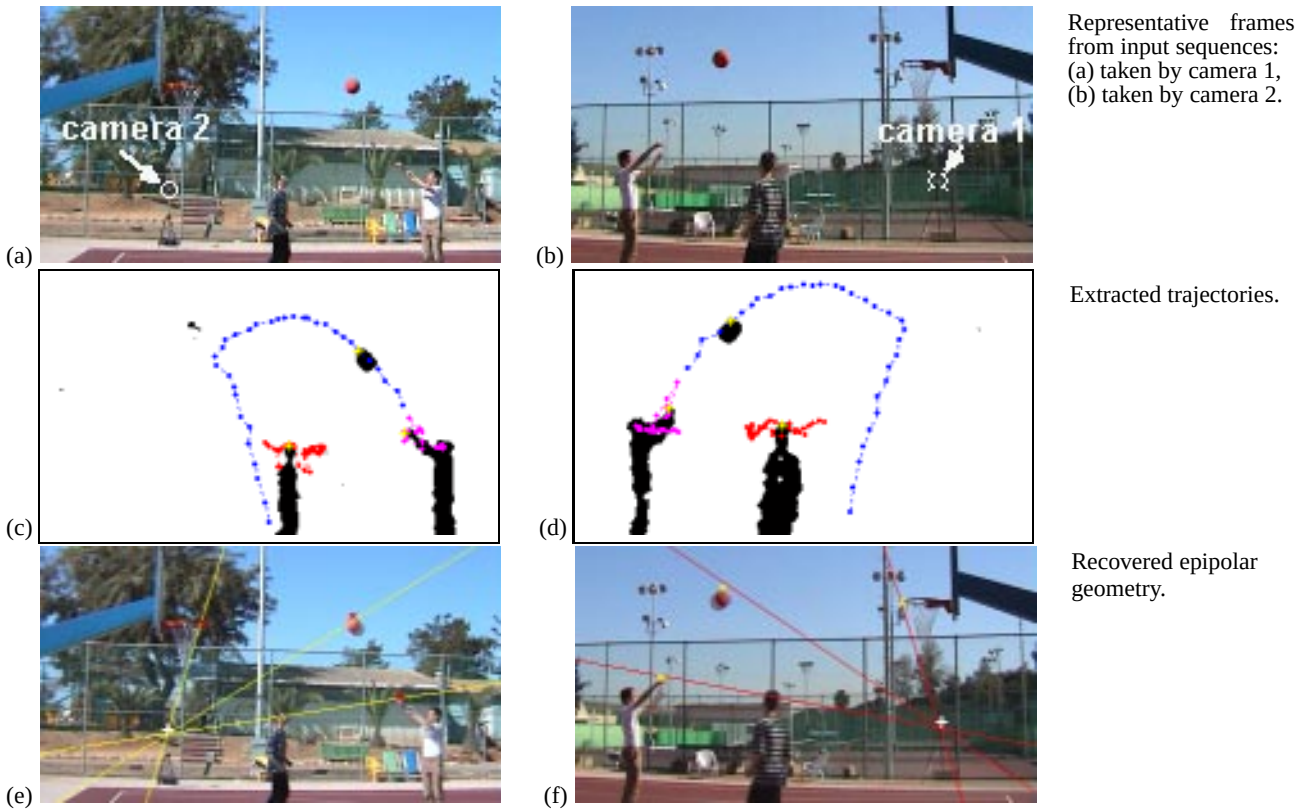



Figure 5: **Wide Base-Line Matching** (see text).
For color sequences see `www.wisdom.weizmann.ac.il/~vision/traj2traj.html`