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.

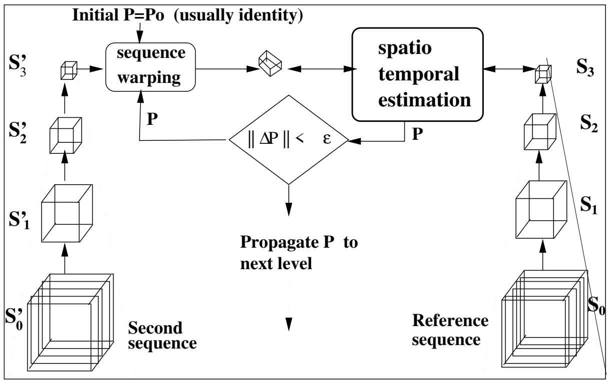


Figure 1. The hierarchical spatio-temporal alignment framework *A volumetric pyramid is constructed for each input sequence, one for the reference sequence (on the right side), and one for the second sequence (on the left side). The spatio-temporal alignment estimator is applied iteratively at each level. 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 estimation 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 3×3 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:

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

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 (δH and δd). 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 256×256 pixels, then the next resolution level contains a sequence of 32 frames of size 128×128, 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.

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].

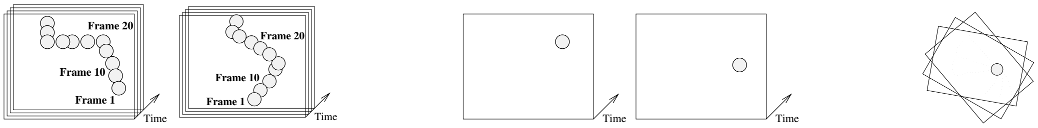
When there are no temporal changes in the scene, all temporal derivatives within the sequence are zero: $S_t \equiv 0$. Therefore, for any voxel (x, y, t), the error term of Eq. (3) reduces to:

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

$$= I' - I + [u,v]\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \underbrace{e_{img}(x,y;\vec{P})}_{\text{img-to-img}}.$$

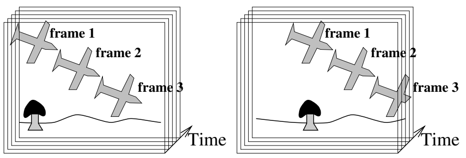
where, $I(x, y) = S(x, y, t)$ is the image frame at time t.

[2] A Laplacian pyramid can equally be used.

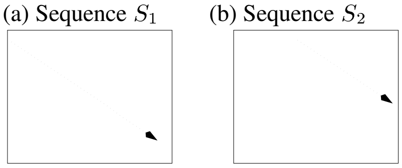


(a) Sequence S_1 (b) Sequence S_2 (c)Frame from S_1 (d)Frame from S_2 (e) Possible image alignments

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

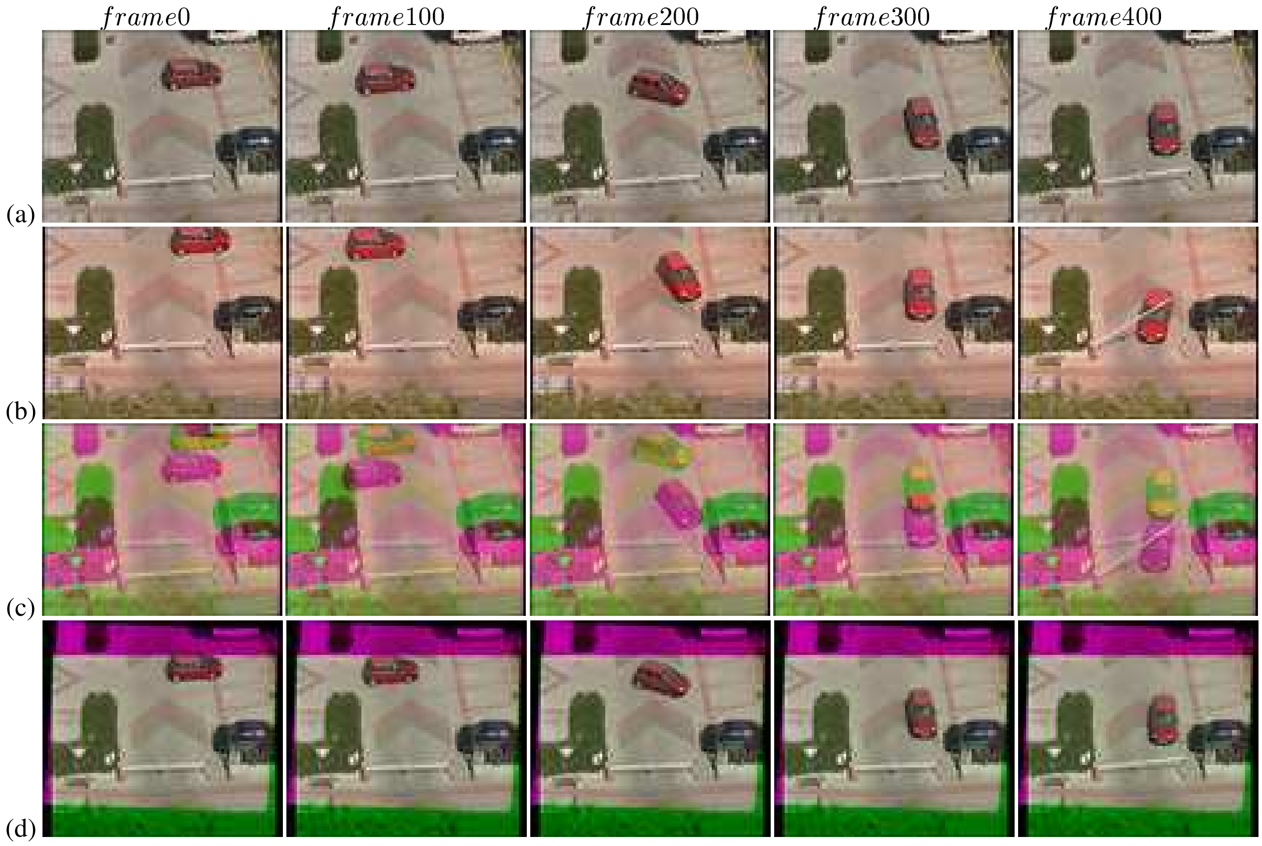


Figure 4. Scene with moving objects. *Rows (a) and (b) display five representative frames (0,100,200,300,400) from the reference and second sequences, respectively. The spatial misalignment is easily observed near image boundaries, where different static objects are visible in each sequence. The temporal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

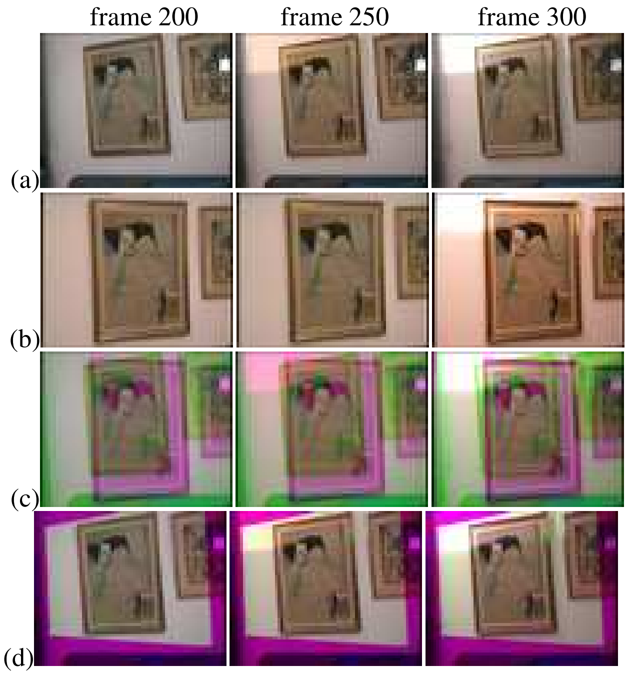


Figure 5. Scene with varying illumination. *Rows (a) and (b) display three representative frames (200,250,300) from the reference and second sequences, respectively. The temporal misalignment can be observed in the upper left corner of frame 250, by small differences in illumination. (c) displays superposition 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 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 Δt 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 ω is the upper bound on the spatial frequencies. Applying this rule to our case, yields the following constraint: $v\Delta t = \Delta trace <= \frac{1}{2\omega}$. This equation characterizes the *temporal* sampling rate which is required to avoid temporal aliasing. In practice, video sequences of scenes with fast moving objects often contain temporal aliasing. We cannot control the frame rate $(\frac{1}{\Delta t})$ nor object's motion (v). We can, however, decrease the spatial frequency upper bound ω by reducing the spatial resolution of each frame (i.e., apply a spatial low-pass-filter). This implies that for video sequences which inherently have high temporal aliasing, it may be necessary to compromise in spatial resolution of alignment in order to obtain correct temporal alignment. Therefore, the LPF (low pass filters) in our spatio-temporal pyramid construction (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.

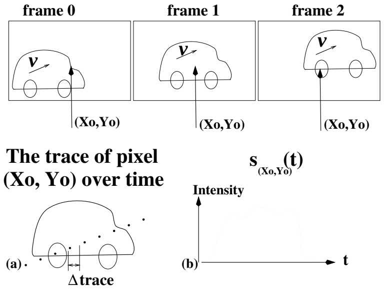


Figure 6. Induced temporal frequencies. *Three frames 0,1,2 of a car moving up right with velocity v are presented above. A fixed pixel (x_0, y_0) is marked on each frame. (a) displays the trace of the pixel. (b) displays the gray level values along this trace.*

Acknowledgment

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