Video Registration Using Dynamic Textures

Avinash Ravichandran, *Student Member*, *IEEE*, and René Vidal, *Member*, *IEEE*

Abstract—We consider the problem of spatially and temporally registering multiple video sequences of dynamical scenes which contain, but are not limited to, nonrigid objects such as fireworks, flags fluttering in the wind, etc., taken from different vantage points. This problem is extremely challenging due to the presence of complex variations in the appearance of such dynamic scenes. In this paper, we propose a simple algorithm for matching such complex scenes. Our algorithm does not require the cameras to be synchronized, and is not based on frame-by-frame or volume-by-volume registration. Instead, we model each video as the output of a linear dynamical system and transform the task of registering the video sequences to that of registering the parameters of the corresponding dynamical models. As these parameters are not uniquely defined, one cannot directly compare them to perform registration. We resolve these ambiguities by jointly identifying the parameters from multiple video sequences, and converting the identified parameters to a canonical form. This reduces the video registration problem to a multiple image registration problem, which can be efficiently solved using existing image matching techniques. We test our algorithm on a wide variety of challenging video sequences and show that it matches the performance of significantly more computationally expensive existing methods.

Index Terms—Dynamic textures, video registration, nonrigid dynamical scenes.

1 Introduction

A classical problem in computer vision is that of aligning images of the same scene taken at different instances. These instances can either be different view points or different time instances. This problem is known as image registration and the objective is to recover the correspondences between the images. Once such correspondences have been found, all of the images can be transformed into the same frame of reference. This enables one to either compare the images or augment the information in one image with the information from the others. Image registration finds a wide variety of applications in computer vision, such as image stitching, image-based modeling and rendering, structure-from-motion, object recognition, etc. Image registration is also important in medical imaging, where multimodal data can be used to augment the information in one image or images taken at different times can be compared to assess the evolution of a disease.

Image registration methods can be broadly divided into two categories—feature-based methods and direct methods. In feature-based methods, feature points such as Harris corners [16], scale invariant feature transform (SIFT) features [19], multiscale-oriented patches (MOPSs) [6], etc., are first extracted from the images. These features are then matched using methods such as normalized cross correlation. Once a rough (possibly incorrect) set of matches is obtained, one can use methods such as random sample consensus (RANSAC) [14] to refine the matches and calculate the transformation between the images. Direct methods, on the other hand, first define a metric, such as the sum of square differences, mutual information [34], etc. The registration problem is then solved by the minimization/maximization of a cost function built from this metric. We refer the reader to [29] and the references therein for a more detailed review of image registration methods.

1.1 Prior Work on Video Registration

The last few years have seen an increasing interest in the problem of video registration. The term video registration in the existing literature has been interchangeably used for two different problems. The first refers to registering frames of a single video sequences to a chosen frame of the video, while the other refers to registering two different video sequences of the same scene.

The task of registering adjacent frames of a video sequence reduces to a standard image registration problem. However, when the scene is nonrigid, this process becomes more complicated. This is because classical constraints such as the brightness constancy constraint are no longer valid. One of the first methods for registering nonrigid dynamical scenes was proposed by Fitzgibbon [15]. This method combines linear dynamical systems (LDSs) with stochastic rigidity to align frames in a video taken by a single moving camera. The work of Vidal and Ravichandran [33] used time varying LDSs and proposed a method to calculate the optical flow of nonrigid scenes viewed by a single moving camera. Doretto and Soatto [13] extended the linear model to jointly learn the appearance, shape, and the motion for such scenes. Rav-Acha et al. [23] proposed a method based on video interpolation. The difference between the predicted image and the incoming image was used to drive the registration process. Agarwala et al. [4] extended the concept of video textures [27] to the panoramic video texture case. Starting from a panning video sequence, a video sequence containing the dynamics of the entire spatial panorama is generated. Similarly, Rav-Acha et al. [22] proposed the concept of dynamosaicing, in which the registration parameters are found as the minimum cut in a 4D graph.

• *The authors are with the Center for Imaging Science, The Johns Hopkins University, 319A Clark Hall, 3400 N. Charles St, Baltimore, MD 21218. E-mail: {avinash, rvidal}@cis.jhu.edu.*

Manuscript received 2 Oct. 2008; revised 31 July 2009; accepted 25 Sept. 2009; published online 1 Mar. 2010.
Recommended for acceptance by A. Torralba.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-10-0663.
Digital Object Identifier no. 10.1109/TPAMI.2010.61.

The task of registering multiple video sequences to each other involves recovering not only the spatial alignment, as in the case of image registration, but also the temporal alignment. This poses additional challenges compared to image registration. If the two sequences were temporally aligned, one could argue that the video registration problem could be reduced to an image registration problem. However, the main challenge is to decide which pair of frames should be registered. To deal with this issue, one could do a frame-by-frame registration and then choose the transformation that gives the least error. Alternatively, one could find the transformation that minimizes the sum of the errors over all the frames. However, these methods are naive and computationally expensive. They also ignore the dynamics of the scene, which could provide us with more information than just using frame-by-frame comparison. When the two sequences are not temporally aligned, we cannot resort to any of the approaches outlined earlier. Instead, the video sequence can be considered as a space-time volume and volume-to-volume registration techniques can be applied using either the entire volume or a collection of subvolumes, or point trajectories. Caspi et al. have a series of papers [7], [8] that address the problem of spatial-temporal alignment. In [7], two algorithms were proposed—a feature-based and a gradient-based. In the feature-based approach, the features are extracted using either a KLT tracker [28] or using centroids of blobs. The alignment problem is posed as an optimization problem, which is solved using the Gauss-Newton method. The gradient-based method, on the other hand, works directly on the intensities rather than tracked features. However, the algorithm relies on similar appearances between the two video sequences. To overcome this in [8], a feature-based approach is proposed. The features used in this paper are the point trajectories, which make the algorithm invariant to appearance changes. In [31], a unified framework is presented combining the work of the two aforementioned papers. The paper also extends the spatial-temporal alignment problem so that the alignment can be performed even if the two sequences are captured at different instances. A similarity measure is proposed for each space-time subvolume. The registration is then obtained by maximizing a similarity measure across all the subvolumes. However, such methods are computationally intensive as they involve an optimal search for both the spatial and temporal registration parameters over the entire video.

1.2 Paper Contributions

In this paper, we propose a unified video registration algorithm for both rigid and nonrigid sequences. We assume that the two video sequences are related by a rigid transformation in space and an offset in time. We reduce the problem of registering two video sequences to that of registering multiple images. A homography is the common choice of spatial transformation when registering images in order to generate panoramas. Hence, we assume that the spatial transformation between these two video sequences is a homography. We wish to point out that we are not restricted to using only the homography as the model of choice, but can use any rigid transformation between two images such as the fundamental matrix. We also assume that there is an overlap between the two video sequences of the phenomenon they are observing, i.e., if one video sequence is observing a moving object (rigid or nonrigid), the other video sequence also contains portions (spatially and temporally) of this moving object. We model the video sequences as the output of an LDS. Over the past few years, several methods for modeling nonrigid dynamical scenes have been proposed [30], [27], [35], [5], [11], [18]. All of these models are generative, i.e., given a finite number of frames of a video sequence or a finite-sized image, these methods can extend the textures to the desired temporal/spatial size using techniques such as graphcuts, dynamic programming, etc. Among these methods, the *dynamic texture* framework [11] is particularly attractive because the parameters of the model can be clearly exploited for solving computer vision problems. The dynamic texture framework has been used in prior work for the purpose of segmentation [12], recognition [26], [9], [32], and calculating optical flow of nonrigid scenes [33]. Although the dynamic texture model is not an appropriate model for the synthesis of rigid scenes, such a model can be used to register rigid sequences. This is primarily because the quality of the model identified from the two video sequences will be the same. Hence, in order to establish correspondences between the two scenes, the information captured in the dynamic texture model is useful. In addition, for rigid scenes, the mean image of the video sequence contains the background and this is also exploited in our framework.

The contributions of the paper are the following:

1. We propose a feature-based method for registering multiple video sequences using the dynamic texture model. Our approach does not use space-time volumes or feature trajectories. Neither does it rely on ad hoc heuristics such as the appearance images. Instead, we extract traditional image-based features from the model parameters for the registration.
2. Our framework does not require the video sequences to be synchronized. We can recover the spatial transformation independent of the temporal alignment between the two video sequences. In addition, when there is no temporal transformation between the two video sequences, we are able to perform the registration more efficiently. This is because we perform feature matching on the model parameters of the video sequence as opposed to every pair of frames. As we show later, our algorithm reduces to extracting feature matches from $(n+1)$ images pairs as opposed to F (number of frames) images pairs as in the case of traditional algorithms.
3. Since the identification of the parameters of the LDS is not unique, we propose a scheme for the joint identification of the parameters of the LDSs that model the video sequences. The proposed method retains the suboptimality of the original identification algorithm of Doretto et al. [11]. However, our algorithm helps resolve the ambiguities in the parameter estimation and also enforces the same dynamics for multiple video sequences. Our algorithm is very simple and relies on using a canonical form.
4. We outline several choices for the canonical form of the parameters of the LDSs and show that the canonical form based on the real Jordan form overcomes some of the issues with other canonical forms.

We outline a method to solve for the transformation that converts the parameters to the canonical form. By using a canonical form, we convert all the parameters into the same basis and this makes comparing the parameters more straightforward. This method is independent of a reference sequence and scales well for an arbitrary number of sequences.

2 Problem Formulation

In this section, we outline our approach for registering multiple video sequences using the dynamic texture framework. In Section 2.1, we briefly review the dynamic texture model. We refer the readers to [11] for more details on this model. Later in Section 2.2, we show that by modifying the parameters of this model, we can describe sequences that are spatially and temporally transformed versions of the original video sequence. This will serve as the basis of our registration algorithm.

2.1 Dynamic Textures Framework

Given a video sequence $\{I(t)\}_{t=1}^{F}$, we model the temporal evolution of its intensities as the output of an LDS. The equations that model the sequence are given by

$$\mathbf{z}(t+1) = A\mathbf{z}(t) + B\mathbf{v}(t), \tag{1}$$

$$I(t) = C^0 + C\mathbf{z}(t) + \mathbf{w}(t). \tag{2}$$

The parameters of this model can be classified into three types, namely, the appearance, dynamics, and noise parameters. The vector $\mathbf{z}(t) \in \mathbb{R}^n$ represents the *hidden state* of the system at time t. Its evolution is controlled by the *dynamics matrix* $A \in \mathbb{R}^{n\times n}$ and the *input-to-state matrix* $B \in \mathbb{R}^{n\times q}$. These parameters are termed the *dynamics parameters* of the dynamic texture model. The parameter $C \in \mathbb{R}^{p\times n}$ maps the hidden state to the image and the vector $C^0 \in \mathbb{R}^p$ is the temporal mean of the video sequence. These parameters are called the *appearance parameters* of the dynamic texture model. The *noise parameters* are given by the zero-mean Gaussian processes $\mathbf{v}(t) \sim \mathcal{N}(0, Q)$ and $\mathbf{w}(t) \sim \mathcal{N}(0, R)$, which model the process noise and the measurement noise, respectively. The order of the system is given by n and p is the number of pixels in the image. The advantage of using this model is that it enables us to decouple the appearance parameters of the video sequence from the dynamics parameters. Thus, if one is interested in recovering appearance-based information from the video sequence, such as optical flow or, in our case, the spatial registration, then one only needs to deal with the appearance parameters. This allows us to recover the spatial registration independent of the temporal alignment of the sequence, as will be seen in the next section.

2.2 Recovering the Spatial-Temporal Transformation from the Parameters of Dynamic Textures

As motivated in the previous section, spatial registration can be recovered using a subset of the parameters of the LDS. In this section, we explore the relationship between the parameters of two video sequences that are spatially and temporally transformed versions of each other.

Let $\mathbf{x} = (x, y)$ be the coordinates of a pixel in the image. We define $C^i(\mathbf{x})$ to be the ith column of the C matrix reshaped as an image. Likewise, we define $C^0(\mathbf{x})$ to be the mean of the video sequence reshaped as an image. With this notation, the dynamic texture model can be rewritten as

$$I(\mathbf{x}, t) = \sum_{i=0}^{n} z^i(t) C^i(\mathbf{x}) + \mathbf{w}(t), \tag{3}$$

where $z^0(t) = 1$. Therefore, under the dynamic texture model, a video sequence is interpreted as an affine combination of n basis images and the mean image. We call these $n+1$ images $\{C^i(\mathbf{x})\}_{i=0}^{n}$ the *dynamic appearance images*.

In the following analysis, we consider a video sequence and its corresponding LDS. We show that a spatial and temporal transformation on the video sequences induces a spatial and temporal transformation in the parameters of the LDS. We first consider the simple case of the two video sequences being just spatially transformed version of each other. Such video sequences are termed as *synchronized* video sequences. We then consider the more general case of two video sequences being both spatially and temporally transformed versions of each other. Such video sequences are also known as *unsynchronized* video sequences.

2.2.1 Synchronized Video Sequences

Let $T\colon \mathbb{R}^2 \to \mathbb{R}^2$ be any spatial transformation relating the frames from each video, such as a 2D affine transformation or a homography. The relationship between two synchronized video sequence is then given by $\tilde{I}(\mathbf{x}, t) = I(T(\mathbf{x}), t)$. Consider now the following LDSs, with the evolution of the hidden state as

$$\mathbf{z}(t+1) = A\mathbf{z}(t) + B\mathbf{v}(t), \tag{4}$$

and the outputs defined as

$$I(\mathbf{x}, t) = \sum_{i=0}^{n} z^i(t) C^i(\mathbf{x}) + \mathbf{w}(t), \tag{5}$$

$$\tilde{I}(\mathbf{x}, t) = \sum_{i=0}^{n} z^i(t) C^i(T(\mathbf{x})) + \mathbf{w}(t). \tag{6}$$

We can see that $\tilde{I}(\mathbf{x}, t) = I(T(\mathbf{x}), t)$. This shows that when a constant spatial transformation is applied to all the frames in a video sequence, the transformed video can be represented by an LDS that has the same A and B matrices as the original video. The main difference is that the dynamic appearance images $\{C^i(\mathbf{x})\}_{i=0}^{n}$ are transformed by *the same* spatial transformation applied to the frames of the sequences, i.e., $\{\tilde{C}^i(\mathbf{x}) = C^i(T(\mathbf{x}))\}_{i=0}^{n}$.

2.2.2 Unsynchronized Video Sequences

In this case, in addition to the spatial transformation, we now introduce a temporal lag between the two video sequences denoted by τ. The relationship between two unsynchronized video sequence can be represented as $\tilde{I}(\mathbf{x}, t) = I(T(\mathbf{x}), t + \tau)$. Now let us consider the following two systems, with the evolution of the hidden states given by (4), and the outputs defined as

$$I(\mathbf{x},t) = \sum_{i=0}^{n} z_i(t) C^i(\mathbf{x}) + \mathbf{w}(t), \tag{7}$$

$$\tilde{I}(\mathbf{x},t) = \sum_{i=0}^{n} z_i(t+\tau) C^i(T(\mathbf{x})) + \mathbf{w}(t+\tau). \tag{8}$$

We now see that the above equations model two unsynchronized sequences. Thus, a video sequence that is a spatially and temporally transformed version of the original video sequence can be represented with an LDS with the same A and the same B as the original video sequences. However, in addition to the C matrix being modified by the spatial transformation, as in the synchronized case, we also have a different initial state. Instead of the video sequence starting at $\mathbf{z}(0)$, the initial state now is $\mathbf{z}(\tau)$. Nevertheless, if one wants to only recover the spatial transformation, the C matrices of the two LDSs are the only parameters that need to be compared.

Thus, given two video sequences, either synchronized or the unsynchronized, in order to recover the spatial registration, we only need to compare the C matrices. But this is under the assumption that both the A matrix and the state of the system $\mathbf{z}(t)$, modulo a temporal shift $\tau_i \in \mathbb{Z}$, for the two systems remain the same. The rationale behind this assumption is that since the video sequences are of the same scene, the evolution of the hidden states remains the same. More specifically, the objects in the scene undergo the same deformation; hence, they possess the same dynamics. However, if one learns the parameters of the LDSs from the data using existing methods, one encounters two problems. The first problem is that the A matrix is not the same for the different LDSs. Second, the C matrices that are recovered are only unique up to an invertible transformation. Hence, in order to perform the registration, we address these issues in the next section.

3 Recovering Transformation Parameters from the Dynamic Texture Model

In the previous section, we have introduced the dynamic texture model and shown how the model parameters vary for video sequences taken at different viewpoints and time instances. In this section, we will show how the spatial transformation can be recovered using the parameters of the LDSs. In Section 3.1, we review the classical system identification algorithm for learning the parameters of an LDS and show that the recovered parameters are not unique. Since we would like to compare the parameters of different LDSs, our first task is to remove such ambiguities. This issue is addressed in Section 3.2. We then, in Section 3.3, show how we can enforce the dynamics of multiple video sequences to be the same. Finally, in Section 3.4, we propose an algorithm to recover the spatial and temporal transformation from the dynamic appearance images of two video sequences.

3.1 Parameter Identification

Given a video sequence $\{I(t)\}_{t=1}^{F}$, the first step is to identify the parameters of the LDS. There are several choices for the identification of such systems from the classical system identification literature, e.g., subspace identification methods such as N4SID [21]. The problem with such methods is that as the size of the output increases, these methods become computationally very expensive. Hence, traditionally, the method of identification for dynamic textures has been a suboptimal solution proposed in [11]. This method is essentially a Principal Component Analysis (PCA) decomposition of the video sequence. Given the video sequence $\{I(t)\}_{t=1}^{F}$, the mean $C^0 = \frac{1}{F}\sum_{t=1}^{F} I(t)$ is first calculated. The parameters of the system are then identified from the compact (rank-n) SVD of the mean subtracted data matrix as

$$[I(1) - C^0, \ldots, I(F) - C^0] = U(SV^\top) = CZ, \tag{9}$$

where $Z = [\mathbf{z}(1) \ldots \mathbf{z}(F)]$. Given Z, the parameter A is obtained as the least-square solution to the system of linear equation $A[\mathbf{z}(1) \ldots \mathbf{z}(F-1)] = [\mathbf{z}(2) \ldots \mathbf{z}(F)]$.

It is well known that the factorization obtained from the SVD is unique up to an invertible transformation, i.e., the factors that are recovered are (CP^{-1}, PZ), where $P \in \mathbb{R}^{n \times n}$ is an arbitrary invertible matrix. Hence, the LDSs with parameters (A, B, C) and (PAP^{-1}, PB, CP^{-1}) both generate the same output process. This fact does not pose a problem when dealing with a single video sequence. However, when one wants to compare the parameters identified from multiple sequences, each set of identified parameters could potentially be computed with respect to a different basis. Since our goal is to compare the C matrices, to perform the registration we need to ensure that different C matrices are in the same basis. In order to address this issue, in the next section, we outline a method to account for the basis change. We propose to do this by using a canonical form and converting the parameters into the canonical form.

3.2 Canonical Forms for Parameter Comparison

Given the parameters of an LDS (A, B, C), the family of parameters that generate the same output process is given by (PAP^{-1}, PB, CP^{-1}). There are several approaches to removing the ambiguities from the system parameters. For instance, one can restrict the columns of the C matrix to be orthogonal. Exploiting this fact, one option to overcome the basis ambiguities is to project all the C matrices into the subspace spanned by one of the C matrices. In [10], Chan and Vasconcelos used such an approach where one sequence was chosen as the reference and the parameters of the other sequence were converted into the basis of the reference sequence. One drawback of such a method is that it requires a reference sequence. Choosing such a reference sequence might not always be feasible. An alternate approach from linear systems theory, to address the basis issue, is to use a canonical form. The advantage of such canonical forms is that the model parameters in the canonical forms have a specific structure. As a consequence, if the model parameters identified using the suboptimal approach are converted to the canonical form, the parameters are in the same basis. This removes the basis ambiguity induced in the suboptimal identification algorithm due to the SVD factorization. Also, using the canonical form does not require a reference sequence.

If one refers to the literature from linear systems theory, several canonical forms have been proposed for the model parameters of an LDS in the particular case of a single output system, i.e., $p = 1$. Although, in principle, any canonical form can be used to overcome the ambiguities, the fact that these LDSs model the temporal evolution of the intensities of pixels of a video sequence poses some constraints in the choice of

the canonical form. For example, we do not want such forms to be complex. This would make it difficult to perform comparison between parameters of different systems. Also, even though, theoretically, all of the canonical forms are equivalent, in practice they differ in their numerical stability. Vidal and Ravichandran in [33] used a diagonal form for the A matrix. Since the equivalence class of parameters for A is PAP^{-1}, i.e., a similarity transformation, the diagonal form reduces to the diagonal matrix of eigenvalues of A. Thus, the resulting parameters in canonical form can be complex, since the eigenvalues of A can be complex. To overcome this, the Reachability Canonical Form (RCF) was used in [24]. The RCF is given by

$$A_c = \left[\begin{array}{c|c} \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix} & I_{n-1} \\ \hline -a_0 & -a_1 \ -a_2 \ -a_3 \ \cdots \ -a_{n-1} \end{array}\right] \in \mathbb{R}^{n\times n} \quad \text{and}$$
$$B_c = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \end{bmatrix}^\top \in \mathbb{R}^{n\times 1}, \tag{10}$$

where $A^n + a_{n-1}A^{n-1} + \cdots + a_0 I = 0$ is the characteristic polynomial of A and I_{n-1} is the identity matrix of size $n-1$. The problem with the RCF is that it uses the pair (A, B) to convert the system into canonical form. For most common applications of dynamic textures, such as registration and recognition, it is preferable to have a canonical form based on the parameters (A, C) because they model the appearance and the dynamics of the system. The matrix B, on the other hand, models the input noise and is not that critical to describe the appearance of the scene. Thus, a suitable candidate for the canonical form is the Observability Canonical Form (OCF) [25] given by

$$A_c = \left[\begin{array}{c|c} \begin{matrix} -a_{n-1} \\ -a_{n-2} \\ \vdots \\ -a_1 \end{matrix} & I_{n-1} \\ \hline -a_0 & 0 \ \ 0 \ \ 0 \ \cdots \ 0 \end{array}\right] \quad \text{and}$$
$$C_c = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{1\times n}. \tag{11}$$

However, the estimation of the transformation that converts a set of parameters to this canonical form is numerically unstable [25]. As a result, in the presence of noise, two dynamical systems that are similar can be mapped to dynamical systems in the canonical form that are fairly different.

In order to address this drawback, we propose to use a canonical form based on the Jordan real form. When A has $2q$ complex eigenvalues and $n-2q$ real eigenvalues, the Jordan Canonical Form is given by

$$A_c = \begin{bmatrix} \sigma_1 & \omega_1 & 0 & \cdots & 0 \\ -\omega_1 & \sigma_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \lambda_{2q-n-1} & 0 \\ 0 & 0 & \ldots & 0 & \lambda_{2q-n} \end{bmatrix} \quad \text{and}$$
$$C_c = [1 \ \ 0 \ \ 1 \ \ 0 \ \ \ldots \ \ 1 \ \ 1], \tag{12}$$

where the eigenvalues of A are given by $\{\sigma_1 \pm i\omega_1, \sigma_2 \pm i\omega_2, \ldots, \sigma_q \pm i\omega_q, \lambda_1, \ldots, \lambda_{n-2q}, \}$. It can be noted that the JCF is indeed equivalent to the RCF or the OCF, but in a different basis.

Given any general canonical form based on A and C, we now outline the steps to convert the identified parameters into the canonical form. Assume that we have the identified parameters (A, C). We now need to find an invertible matrix P such that $(PAP^{-1}, \gamma^\top CP^{-1}) = (A_c, C_c)$, where the subscript c represents any canonical form. The vector $\gamma \in \mathbb{R}^p$ is an arbitrary vector chosen to convert the LDS (A, C) with p outputs to a canonical form, which is defined for only one output. In our experiments, we set $\gamma = [1 \ \ 1 \ldots 1]^\top$ so that all rows of C are weighted equally. The relation between the A matrix and its canonical form A_c is a special form of the Sylvester equation:

$$A_c P - PA = 0. \tag{13}$$

Vectorizing this equation, we can solve for P as

$$\text{vec}(P) = \text{null}(I \otimes A_c - A^\top \otimes I), \tag{14}$$

where $\otimes$ represents the Kronecker product. Similarly, if we consider the equation between the C matrices, $C_c P = \gamma^\top C$, and vectorize it, we can solve for P by concatenating the two sets of equations as follows:

$$\begin{bmatrix} I \otimes A_c - A^\top \otimes I \\ I \otimes C_c \end{bmatrix} \text{vec}(P) = \begin{bmatrix} 0 \\ C \end{bmatrix}. \tag{15}$$

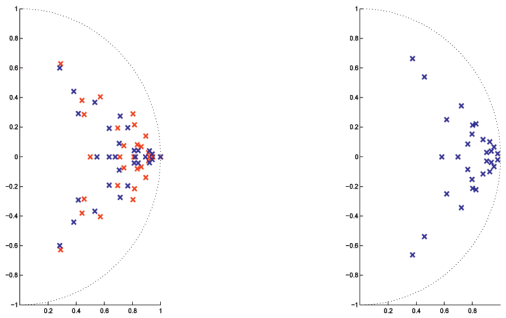
Once we have solved this equation, we can convert the parameters into the canonical form using P. It should be noted that the JCF is unique only up to a permutation of the eigenvalues. However, if we select a predefined order to sort the eigenvalues, we obtain a unique JCF.

3.3 Joint Identification of Dynamic Textures

In the prior section, we have shown how to convert the identified parameters into the same basis so that we can compare the C matrices to recover the registration. However, comparing the C matrices to recover the spatial transformation is based on using the assumption that the A matrices for the two systems were the same. This assumption is valid as we observe the same scene, and since the hidden states $\mathbf{z}(t)$ captures the scene dynamics, they must evolve in the same way, irrespective of the viewpoint.

However, if we identify the LDS from each video sequence separately due to the presence of noise, viewpoint changes, and the suboptimal identification, there is no guarantee that the A matrix for the video sequences will be the same. This can be seen in Fig. 1a. We see that the eigenvalues of the A matrices identified separately are not the same. However, they are close to each other.

In this section, we propose a simple method to explicitly enforce the dynamics of multiple LDSs to be the same.



(a) (b)

Fig. 1. Eigenvalues of the identified A matrices of two spatially and temporally transformed video sequences. The red crosses denote the eigenvalues for sequence 1 and the blue crosses denote the eigenvalues of sequence 2. (a) Separate identification. (b) Joint identification.

Consider M video sequences, each represented as $I_i(t) \in \mathbb{R}^{p_i}, t \in \{1 \dots F\}, i \in \{1 \dots M\}$. Let us introduce $\tilde{I}_i(t) = I_i(t) - C_i^0$ for notational brevity. The traditional identification works by first forming the matrix $W_i = [\tilde{I}_i(1) \dots \tilde{I}_i(F)]$, and then, calculating the singular value decomposition of $W_i = U_i S_i V_i^\top$. The parameters of the LDS are identified as $C_i = U_i(:, 1:n)$ and $Z_i = S_i(1:n, 1:n)V_i(:, 1:n)^\top$. In our approach, we instead stack all the videos to form a single W matrix and factorize it using the SVD as

$$W = \begin{bmatrix} \tilde{I}_1(1) & \dots & \tilde{I}_1(F) \\ \vdots & & \\ \tilde{I}_M(1) & \dots & \tilde{I}_M(F) \end{bmatrix} = USV^\top. \tag{16}$$

Although this seems to be the intuitively obvious thing to do, we will now show that this is indeed the correct thing to do. If, for the sake of analysis, we ignore the noise terms, we obtain the state evolution as $\mathbf{z}(t) = A^t \mathbf{z}_0$, where $\mathbf{z}_0$ is the initial state of the system. Now if we consider the temporal lag $\tau_i \in \mathbb{Z}$ for the ith video sequences, then the evolution of the hidden state of the ith sequence is given by $\mathbf{z}_i(t) = A^{\tau_i}\mathbf{z}(t)$. Therefore, we can now decompose W using the SVD as follows:

$$\begin{aligned} W &= \begin{bmatrix} C_1 A^{\tau_1}\mathbf{z}(1) & \dots & C_1 A^{\tau_1}\mathbf{z}(F) \\ \vdots & & \\ C_M A^{\tau_M}\mathbf{z}(1) & \dots & C_M A^{\tau_M}\mathbf{z}(F) \end{bmatrix} \\ &= \begin{bmatrix} C_1 A^{\tau_1} \\ \vdots \\ C_M A^{\tau_M} \end{bmatrix} [\mathbf{z}(1) \dots \mathbf{z}(F)] = CZ. \end{aligned} \tag{17}$$

From the above equation, we can estimate a single common state for all the sequences. Moreover, given Z, we estimate a common dynamics matrix for all the sequences. Now we can also recover C_i from C up to the matrix A^{τ_i}. The problem is that τ_i is unknown, so we cannot directly compute C_i from C. Now if we consider the equation for the ith video sequence, we can see that

$$[\tilde{I}_i(1) \dots \tilde{I}_i(F)] = C_i A^{\tau_i}.[\mathbf{z}(1) \cdots \mathbf{z}(F)] \tag{20}$$

$$= C_i A^{\tau_i} (A^{\tau_i})^{-1} [\mathbf{z}(\tau_i + 1) \cdots \mathbf{z}(F + \tau_i)]. \tag{21}$$

Thus, we see that the parameters we estimate are the original parameters of the system, but in a different basis. Therefore, by converting the parameters to the canonical form, we can remove the trailing A^{τ_i} and recover the original parameters in their canonical form. The joint identification algorithm is outlined in Algorithm 1. Now, by construction, the A matrices of the multiple video sequences are the same. Hence, their eigenvalues are also the same. This can be seen in Fig. 1b.

Algorithm 1. Joint identification of video sequences

1 Given m video sequence $\{I_i(t) \in \mathbb{R}^{p_i}\}_{i=1}^m$, calculate the temporal mean of each sequence $C_i^0 \in \mathbb{R}^{p_i}$ and set $\tilde{I}_i(t) = I_i(t) - C_i^0$.

2 Compute C, Z using the rank n singular value decomposition of the matrix

$$W = \begin{bmatrix} \tilde{I}_1(1), & \dots & , \tilde{I}_1(F) \\ \vdots & & \\ \tilde{I}_m(1), & \dots & , \tilde{I}_m(F) \end{bmatrix} = USV^\top, \tag{18}$$

$$Z = SV^\top, \quad C = U \tag{19}.$$

3 Compute $A = [\mathbf{z}(2), \dots, \mathbf{z}(F)][\mathbf{z}(1), \dots, \mathbf{z}(F-1)]^\dagger \in \mathbb{R}^{n \times n}$.

4 Let $C_i \in \mathbb{R}^{p_i \times n}$ be the matrix formed by rows $\sum_{j=1}^{i-1} p_j + 1$ to $\sum_{j=1}^{i} p_j$ of C, and convert the pair (A, C_i) to Jordan canonical form.

Having identified a dynamic texture model for all video sequences with a common A and all C matrices with respect to the same basis, in the next section we describe a method to register multiple video sequences using the appearance parameters of the LDSs. Other applications of using the joint identification include recognition of dynamic textures, joint synthesis of videos, etc.

3.4 Registering Using the Dynamic Texture model

In this section, we propose an algorithm to recover the spatial transformation from the appearance parameters of the LDSs identified from the two video sequences. As elaborated in Section 2.2, the C matrix of the LDS captures the appearance. Hence, we need to compare this parameter between two LDSs, to recover the relative spatial alignment. In addition, the mean of the video sequence also contains information that can be exploited to recover the spatial alignment. In our paper, we term the mean image (C^0) and the n columns of the C matrix as the *dynamic appearance images*. Let us consider two video sequences, $I_1(\mathbf{x}, t)$ and $I_2(\mathbf{x}, t)$, where $\mathbf{x}$ denotes the pixel coordinates and $t = 1, \dots, F$. We assume that the video sequences are related by a Homography H and a temporal lag τ, i.e., $I_1(\mathbf{x}, t) = I_2(H(\mathbf{x}), t + \tau)$. Once we recover the spatial alignment independent of the temporal lag between the video sequences, we temporally align the two sequences using a simple line search in the temporal direction, i.e., $\tau = \text{argmin}_\tau \sum_t \|I_1(\mathbf{x}, t) - I_2(H(\mathbf{x}), t + \tau)\|^2, \tau \in \mathbb{Z}$.

Our algorithm to spatially register the two video sequences $I_1(t)$ and $I_2(t)$ proceeds as follows: We calculate the mean images C_1^0 and C_2^0, identify the system parameters (A, C_1) and (A, C_2) in the JCF, and convert every column of C_i into its image form. We use the notation C_j^i to denote the ith column of the jth sequence represented as an image. We use a feature-based approach to spatially register the two

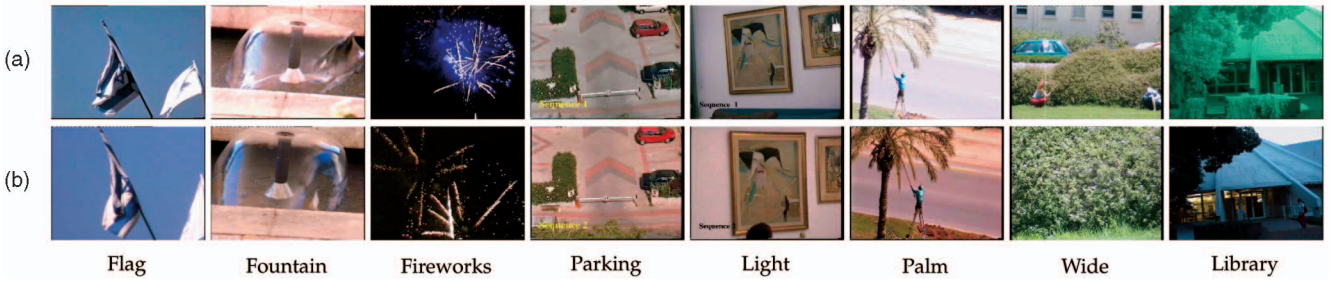


Fig. 2. The first frame from the set of sequences we use for testing the registration algorithm. The first row shows the first frame of Sequence 1 and the second row shows the first frame of the sequence 2 from each video sequence set.

sets of images $\{C_1^0, C_1^1, \ldots, C_1^n\}$ and $\{C_2^0, C_2^1 \ldots, C_2^n\}$. We extract SIFT features and a feature descriptor around every feature point in the two sets of $n+1$ images. We match the features extracted from image C_1^i with those extracted from image C_2^i, where $i \in \{0, \ldots, n\}$, i.e., the forward direction. We also match the features from C_2^i with those extracted from image C_1^i, where $i \in \{0, \ldots, n\}$, i.e., the reverse direction. We retain only the matches that are consistent both in the forward direction and the reverse direction. We then concatenate the correspondences into the matrices $X_1 \in \mathbb{R}^{3\times M}$ and $X_2 \in \mathbb{R}^{3\times M}$. The corresponding columns of X_1 and X_2 are the location of the matched features in homogenous coordinates and M is the total number of matches from the $n+1$ image pairs. We then need to recover a homography H such that $X_2 \sim HX_1$. In order to recover the homography, we first run RANSAC and obtain the inliers from the matches. We then fit a homography using the nonlinear method outlined in [17]. Our registration algorithm is summarized in Algorithm 2.

Algorithm 2. Registration of video sequences

1 Given $I_1(t)$ and $I_2(t)$, calculate the parameters A, C_i^0, and C_i.
2 Extract features and the descriptors from (C_j^i), $j = \{1, 2\}, i = 0, \ldots, n$.
3 Match features from C_1^i to C_2^i and also in the reverse direction. Retain the matches that are consistent across both directions and concatenate the feature point location from C_1^i into X_1 and its corresponding match into X_2
4 Recover the homography H using RANSAC such that $X_2 \sim HX_1$.
5 Calculate temporal alignment τ as $\tau = \arg\min_\tau \sum_t \|I_1(\mathbf{x}, t) - I_2(H(\mathbf{x}), t+\tau)\|^2$.

Remark. For rigid sequences, one could potentially argue that the mean image would be sufficient to register the video sequences. This motivates the fact that for such sequences, we could use only the matches from the mean images rather than the matches from C matrix when estimating the registration parameters. The scenario for nonrigid scenes is the exact opposite. We would like to use the matches from C matrix rather than the matches from the mean image. Note that in our algorithm, the best matches given by RANSAC could arise from the mean image or the dynamic appearance images or both. Hence, we do not explicitly restrict the algorithm to use only the mean image or only the dynamic appearance images, as done in [24]. This choice now becomes automatic and makes our method applicable to both rigid and nonrigid sequences.

4 Experimental Results

In this section, we present an experimental evaluation of the various aspects of our algorithm. We first evaluate the effects of the choice of the canonical forms on the quality of the registration. Since we have two choices, both for our canonical form and for the method of identifying the parameters of the LDSs, we obtain four variations of our algorithm. We can thus obtain the system parameters of the two video sequences using the joint identification (JID) using the JCF, or JID using the RCF, or separate identification (SID) using the JCF, or SID using the RCF. We analyze the performance of these variations of the algorithm with respect to different criteria. We finally present results on real sequences and compare them to existing video registration methods.

In order to evaluate our algorithm, we compiled a database from video sequences used in existing work [7], [8], [31]. The video sequences are available at [1], [2], [3]. This gave us a total set of eight sets of sequences, each set containing two video sequences. Among these, three sets of the sequences were of rigid bodies, namely, the parking sequence, the palm sequences, and the light sequence. Three other sets of sequences were of nonrigid objects, namely, the flag, the fountain, and the fireworks sequences. We also had one pair of sequences with a relatively large zoom factor difference, which we call the wide sequence, and the last set of sequences was from two nonstationary cameras capturing different modalities, which we call the library sequence. Fig. 2 shows sample frames from all the sequences.

4.1 Evaluation of Canonical Forms

To evaluate the effects of the canonical forms on the registration performance, we first take a video sequence and identify the parameters of the system using the suboptimal approach. We then apply different transformations to the system parameters (C, A) to obtain new parameters $(\tilde{C}, \tilde{A}) = (CP^{-1}, PAP^{-1})$. This simulates the ambiguities we encounter when identifying an LDS for each video sequence separately. We then convert (C, A) and $(\tilde{C}, \tilde{A})$ to their canonical form (C_c, A_c) and $(\tilde{C}_c, \tilde{A}_c)$, respectively. Errors between the parameters before and after converting it to the canonical form were calculated using the Frobenius norm $\|\cdot\|_F$. We define the errors as $E_A = \|A - \tilde{A}\|_F$ and $E_C = \|C - \tilde{C}\|_F$. We perform the experiments for 200 random choices of the transformation P and calculate the mean error. The transformations are

TABLE 1
Parameter Errors Before and After Converting to the Canonical Form

Transformation	Initial Error		Errors using RCF		Errors using JCF	
	E_A	E_C	E_A	E_C	E_A	E_C
Sign flip	2.568	7.063	0	0	0	0
Orthogonal	2.61e+00	7.06e+00	1.48e+07	6.46e+08	8.04e-14	1.31e-08
Invertible	1.16e+02	1.51e+02	2.00e+06	1.08e+08	6.79e-10	1.31e-04

Fig. 3. Sample frames of the parking video sequence: Each column shows the corresponding frame from each of the video sequences of the parking set.

randomly generated from three different classes of transformations: a sign flip, an orthogonal matrix, or an invertible matrix. By a sign flip transformation, we refer to a diagonal matrix with entries in $\{-1, 1\}$. The results of this experiment are summarized in Table 1. We see that for the simple transformation such as the sign flip, both the canonical forms perform very well: The errors after converting the LDS into the canonical forms are zero. However, when the transformations get more involved, we see that the errors from the RCF are higher than the initial errors, while the JCF is still able to perform well. In order to qualitatively show the difference between the canonical forms, we compare sample columns from the C matrix of two video sequences from the parking set, which we will later use to show registration results. Fig. 3 shows a few frames from the two video sequences of this set. In Fig. 4a, we present these columns of C matrix identified using SID and the RCF. In Fig. 4b, we show the same result, but by using SID and the JCF. In Fig. 4c, the results are shown using JID and the RCF. Finally, Fig. 4d shows the results using the algorithm proposed in this paper, namely, JID using the JCF. One can see that the corresponding column images for the proposed methods are spatially transformed versions of each other. We, however, note no such spatial correspondence between the C matrices of the two sequences when using SID and the RCF. Note that when using the SID and the JCF, the spatial correspondence can be seen for some basis images, but for others, we notice either little or no correspondences. Although using JID and the RCF exhibits stronger spatial correspondences than SID with the JCF, the quality of the feature matches is better using the JID and the JCF.

4.2 Quantitative Evaluation on Synthetic Sequences

In this section, we perform a comprehensive evaluation of the four variations of the proposed registration algorithm. We performed the analysis on one sequence from four different data sets, namely, the flag, fountain, fireworks, and parking data sets. These video sequences are sorted in increasing order of the rigidity of the sequence. The flag sequence is the most nonrigid, while the parking sequences is the most rigid.

For the synthetic transformation, we first rotate the sequence $I(\mathbf{x}, t)$ counterclockwise and clockwise by an angle of θ to obtain $I_1(\mathbf{x}, t)$ and $\tilde{I}_2(\mathbf{x}, t)$, respectively. This gives us a transformation of 2θ between these two video sequences. In this way, both of the video sequences have identical interpolation artifacts. We then temporally shift $\tilde{I}_2(\mathbf{x}, t)$ by $\tau = 25$ frames to obtain $I_2(\mathbf{x}, t)$. Thus, the relation between $I_1(\mathbf{x}, t)$ and $I_2(\mathbf{x}, t)$ is given by $I_2(\mathbf{x}, t) = I_1(R(2\theta, \mathbf{x}), t + \tau)$. The angle of rotation was chosen such that $\theta \in \{2, 4, 6, 8, 10, 12, 14, 16, 18\}$. We used 100 frames of both sequences. The video sequences are then registered using the four different variations of the algorithm. Since the mean images are common to all the four variations, we analyze the registration only using the features from the n dynamic appearance images. This enables us to judge the performance of different dynamic appearance images better.

The choice of the canonical form and the identification method both influence the dynamic appearance images. Hence, for the different variations of our algorithm, the dynamic appearance images will be different. Consequently, the number of feature extracted and the quality of the matches are dependent on the dynamic appearance images. The quality of the feature matches, in turn, will affect the registration accuracy and the number of inliers. To understand the influence of the various dynamic appearance images, we analyze the performance of the different methods. We define four metrics: the number of feature matches, the number of inliers, the quality of the extracted features, and finally, the registration error.

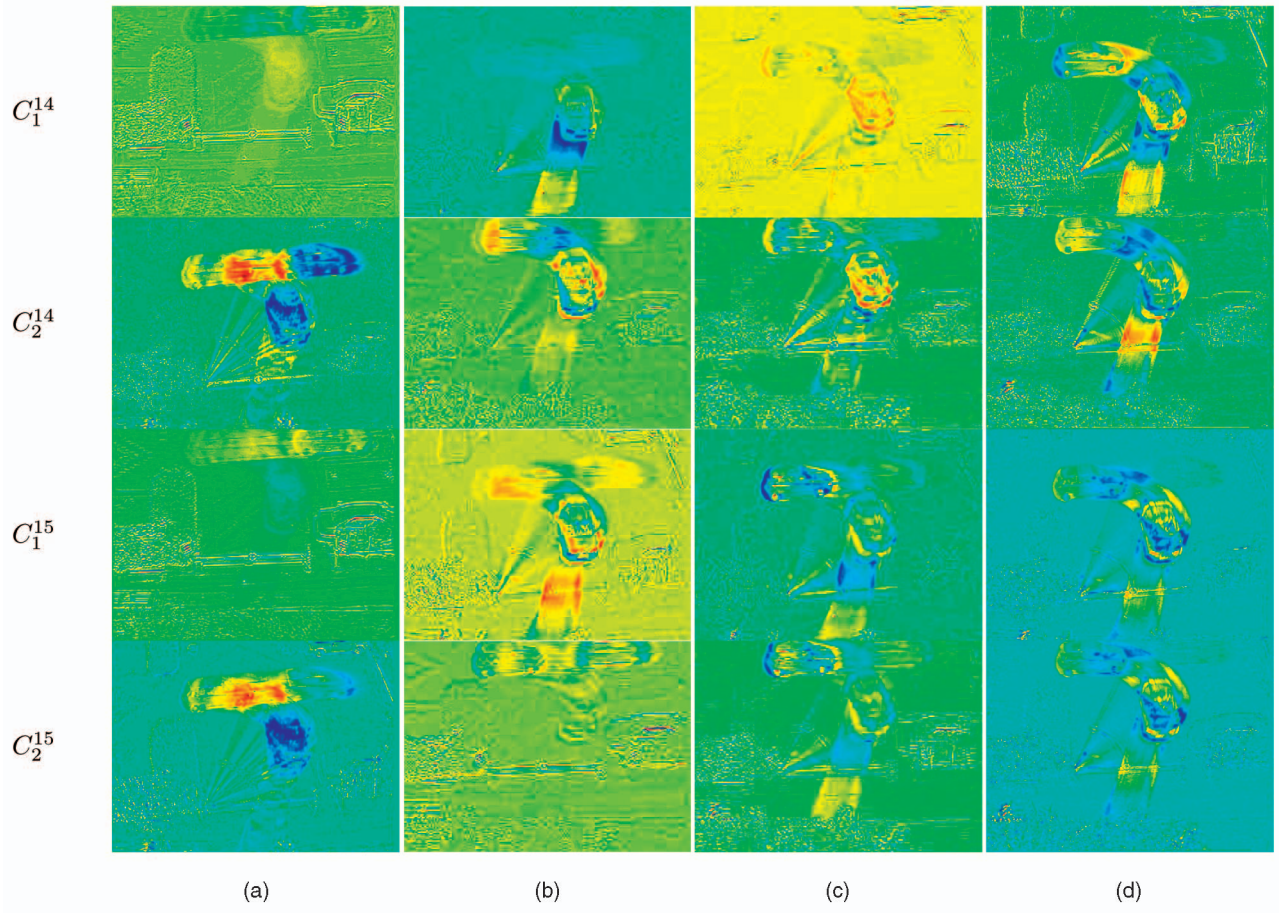


Fig. 4. Sample columns of the C matrix obtained using the different methods for the video sequences from the parking set: Each column shows one variation of our algorithm. C_i^j represents the jth column of the C matrix from the ith video. (a) $\mathrm{SID} + \mathrm{RCF}$. (b) $\mathrm{SID} + \mathrm{JCF}$. (c) $\mathrm{JID} + \mathrm{RCF}$. (d) $\mathrm{JID} + \mathrm{JCF}$.

Fig. 5 shows the number of feature matches extracted from different sequences for varying amounts of synthetic transformation. From this figure, we see that for the flag and the fountain sequences, the greatest number of features is extracted on the dynamic appearance images obtained using JID and the JCF, while for the firework sequence, SID using the JCF results in an equal number of features. For the parking sequence, except for two cases, the greatest number of features is extracted for dynamic appearance images obtained using JID and the RCF. We thus conclude that, except for a few cases, using JID increases the number of feature matches compared to using SID. Also, the JCF consistently gives more matches than the RCF, both while using SID and while useing JID.

We next compare the number of inliers. Fig. 6 shows the number of inliers from different sequences for varying amount of synthetic transformation. For the nonrigid sequences, we observe that using the JCF gives the largest amount of inliers, while for the rigid sequence, JID using the RCF seems to extract more inliers. This trend is anticipated, since, for rigid sequences, the LDS, which is meant to describe the nonrigidity, plays a less significant role in describing the video sequence.

We next consider the quality of the features. The metric we define for the quality of the features is the symmetric homographic transfer error. Given a point correspondence $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2 \in \mathbb{P}^2$ in homogenous coordinates and the recovered homography $H \in \mathbb{R}^{3\times 3}$, the symmetric homographic transfer error is given by

$$E(\mathbf{x}_1, \mathbf{x}_2, H) = \left\| \mathbf{x}_1 - \frac{H^{-1}\mathbf{x}_2}{e_3^\top H^{-1}\mathbf{x}_2} \right\|^2 + \left\| \mathbf{x}_2 - \frac{H\mathbf{x}_1}{e_3^\top H\mathbf{x}_1} \right\|^2. \quad (22)$$

Fig. 7 shows the combined plot for all the video sequences and transformations. Here, the plot of $\log(E)$ is shown versus the cumulative percentage of features. From this figure, we see that using JID with JCF gives us the most reliable features. Given any error, we see that JID using the JCF has the highest percentage of features less than this error, while SID using the JCF follows next.

Our final metric of comparison is the error between the recovered rotation and the ground truth. Although the transformation we recover is a homography ($H = R + TN^\top$), we first decompose the homography into the rotation matrix (R), the normal vector (N), and the scaled translation (T). This gives us two solutions. By using the chirality constraint, we reduce it to one solution. We refer the readers to [20] for details on this decomposition. Note that in this case, we have assumed that the camera is calibrated. We are able to make this assumption here as we apply the rotation to the image and not on the camera. Hence, the intrinsic calibration parameters remain the same for both of the views and can be assumed to be the identity matrix. The error between the true rotation R_t and the recovered rotation R_r is calculated using the distance

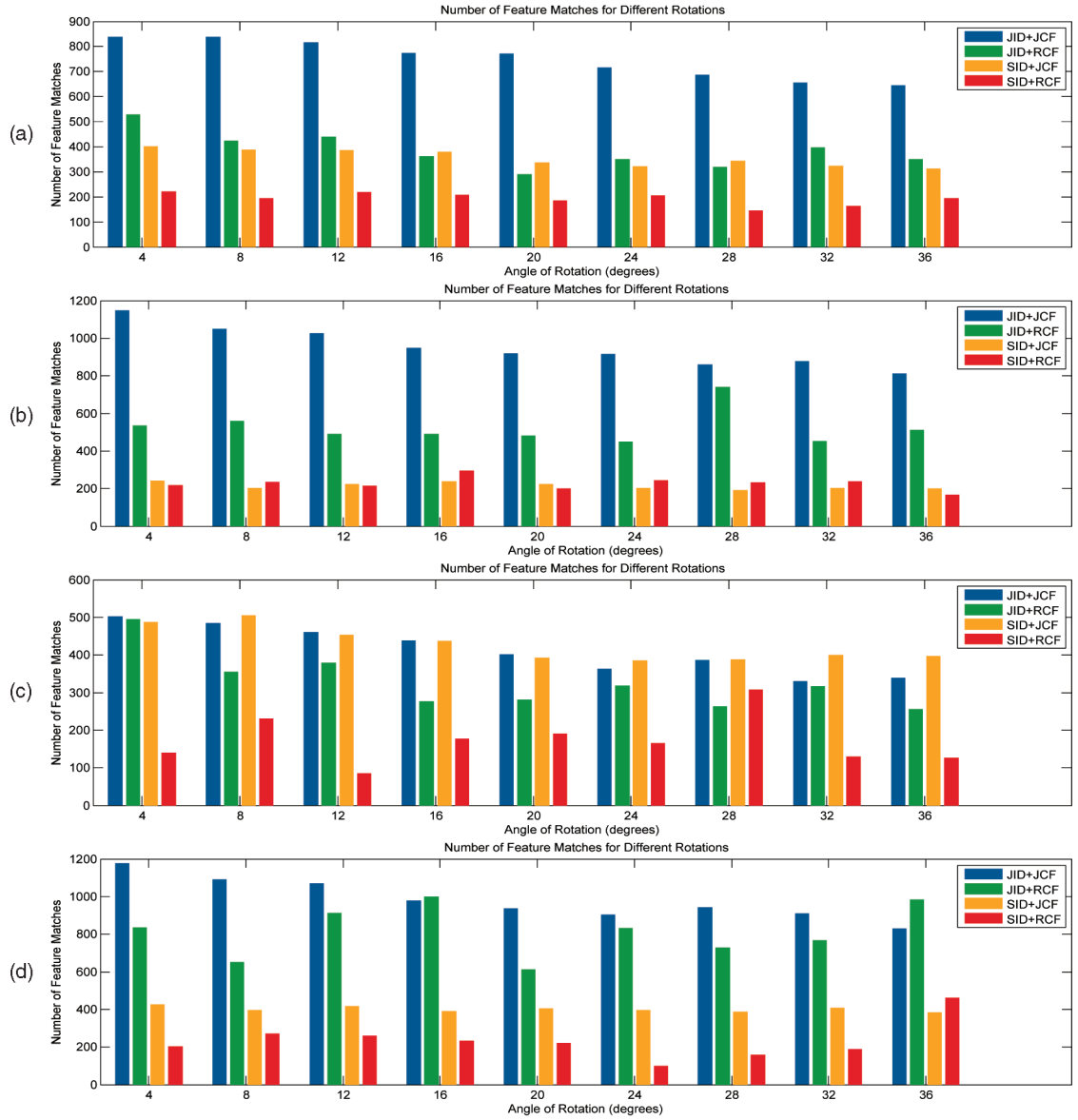


Fig. 5. Comparison of the number of features matches for four different methods. (a) Flag sequence. (b) Fountain sequence. (c) Fireworks sequence. (d) Parking sequence.

$d(R_t, R_r) = \text{acos}(\text{trace}(R_r^\top R_t) - 1)/2)$. The average errors between the true and the recovered transformation are summarized in Table 2. We see that the best errors are obtained using JID and the JCF.

For the metrics we have used in our analysis, we can now rank the different methods based on them either maximizing/minimizing the metric. For example, if we consider the number of matches extracted, we would like the proposed variation of our algorithm to extract as many features as possible. Hence, we assign a rank of one to that variation of the algorithm that extracts the highest matches for a particular transformation of a sequence. The rest are arranged in decreasing order of the number of matches and assigned the ranks of two through four. The same can be done with the number of inliers. But, for the registration error, the ordering needs to be reversed. We then calculate the mean rank of a metric for a sequence and also the overall mean rank across sequences. This result can be seen in Table 3. From this table, we see that for every metric, using JID and the JCF has the best overall mean rank across sequences and transformations, although, for a given sequence, there is no guarantee that using the JID and the JCF is the best with respect to all the metrics. Nevertheless, using JID with the JCF offers us a method that performs consistently across a large variety of sequences.

4.3 Qualitative Evaluation on Real Sequences

For our experiments on real sequences, we tested our algorithm on all of the sequences available in [1]. We also tested our algorithm on one sequence each from [2] and [3] in order to exhibit the variability of the video sequences our registration algorithm can handle. We used an order of $n = 30$ for all of the video sequences shown in this section. In order to qualitatively assess the performance of the registration, we form a new image in which the red and the blue channels come from the first image. The green color channel comes from the green channel of the second

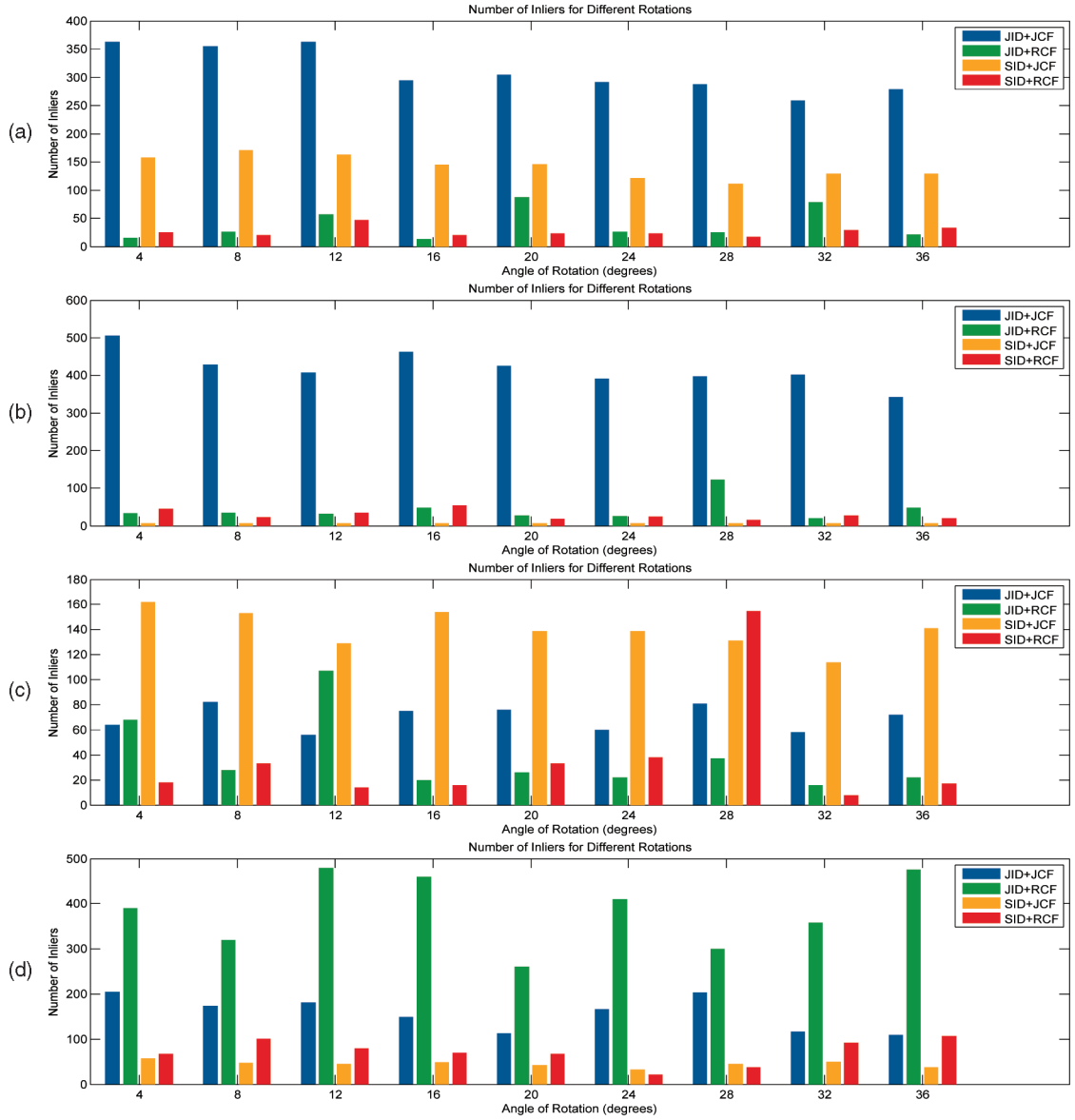


Fig. 6. Comparison of the number of inliers for four different methods. (a) Flag sequence. (b) Fountain sequence. (c) Fireworks sequence. (d) Parking sequence.

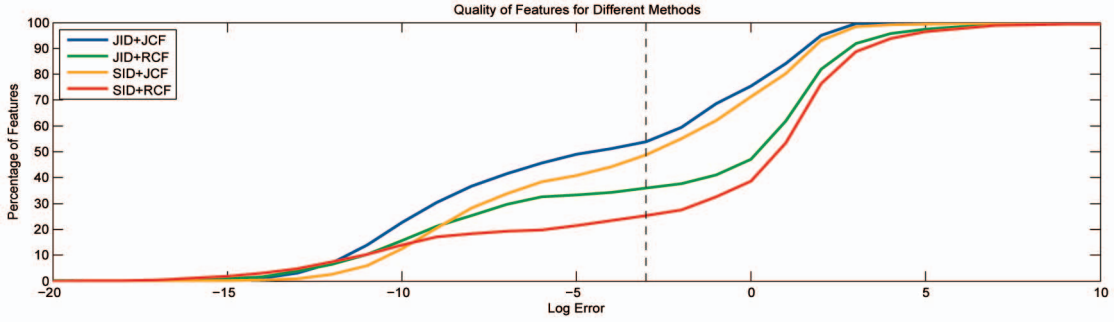


Fig. 7. Comparison of quality of features used by four different methods. The dashed line shows the error threshold for deciding if a feature is an inlier.

image. We show this image both before and after the registration to see the initial alignment and the final result.

We first compare the common sequences from [24] and [7] with our method. We see in Fig. 8 that for all three sequences, the alignment we obtain is as good, if not better, when compared to the other method. We now show additional results in Fig. 9. The sequences here exhibit different kinds of variations such as variation in intensity, shape (nonrigid objects), and modality. We see that in all of the cases, we perform as well as the existing results. We are

TABLE 2
Mean Registration Error over Different Angles for Synthetic Transformations of Sequences

Canonical Form	Identification Method	Mean Error (in radians)			
		Flag	Fountain	Fireworks	Parking
JCF	Joint	0.002	0.001	0.033	0.024
	Separate	0.012	1.999	0.014	0.134
RCF	Joint	2.384	1.792	1.946	0.126
	Separate	2.716	1.791	1.820	1.884

able to register images with a large baseline transformation, as shown in Fig. 9d. Although we have not taken any explicit measure to account for multimodality, we found that, using SIFT features, we are able to register multimodal video sequences, as shown in Fig. 9e. Here, one sequence is captured using a normal camera and the other is captured using an infrared camera. For this case, one can note that our result and the result from the original algorithm do not look the same. This is because we do not perform any kind of fusion. In order to further analyze our algorithm, we obtained the inliers from RANSAC and then calculated the percentage of inliers from the mean features and the rest of the dynamic appearance images. This result can be seen in Table 4. The interesting fact about these numbers is that we see that the algorithm adapts itself based on the sequences. We see that the percentage of inliers from the dynamic appearance images is 100 percent for the flag sequence. This is in agreement with the fact that this is our most nonrigid sequence. Thus, we see that the algorithm performs very well on a large variety of sequences and the results we obtained are comparable to existing methods. The videos of these results can be found at http://vision.jhu.edu/papers/DTReg/.

5 Discussion and Conclusion

We have proposed a method for registering video sequences based on the dynamic texture model. As compared to [7], we are able to recover the spatial transformation independent of the temporal transformation. Our results show that our method performs equivalently to theirs. However, our

TABLE 3
Mean Rank for Different Criterion

Sequence	Registration Error				Number of Inliers				Number of Features			
	JID + JCF	JID + RCF	SID + JCF	SID + RCF	JID + JCF	JID + RCF	SID + JCF	SID + RCF	JID + JCF	JID + RCF	SID + JCF	SID + RCF
Flag	1.1	3.4	1.9	3.6	1.0	3.3	2.0	3.7	1.0	2.3	2.7	4.0
Fountain	1.0	3.0	3.0	3.0	1.0	2.4	4.0	2.6	1.0	2.0	3.6	3.4
Fireworks	2.0	3.4	1.4	3.1	2.3	3.2	1.1	3.3	1.6	3.0	1.6	3.9
Parking	1.4	2.2	2.4	3.9	2.0	1.0	3.8	3.2	1.2	1.8	3.1	3.9
Overall	1.4	3.0	2.2	3.4	1.6	2.5	2.7	3.2	1.2	2.3	2.7	3.8

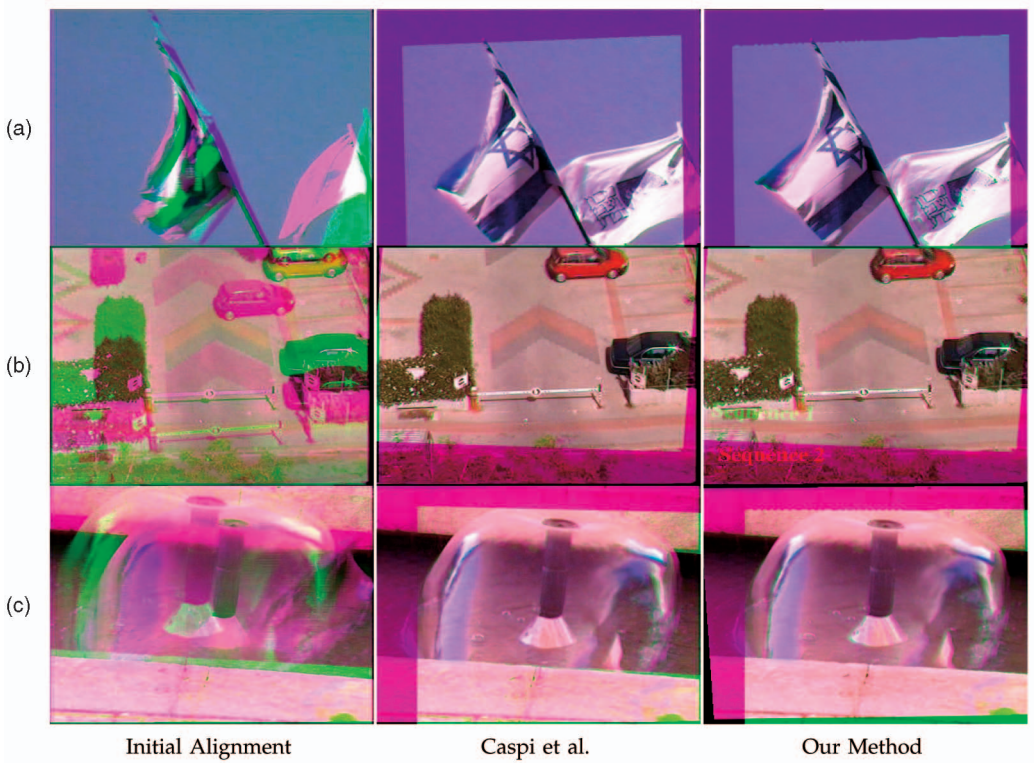


Fig. 8. Comparison of results from two methods. (a) Flag sequence. (b) Parking sequence. (c) Fountain sequence.

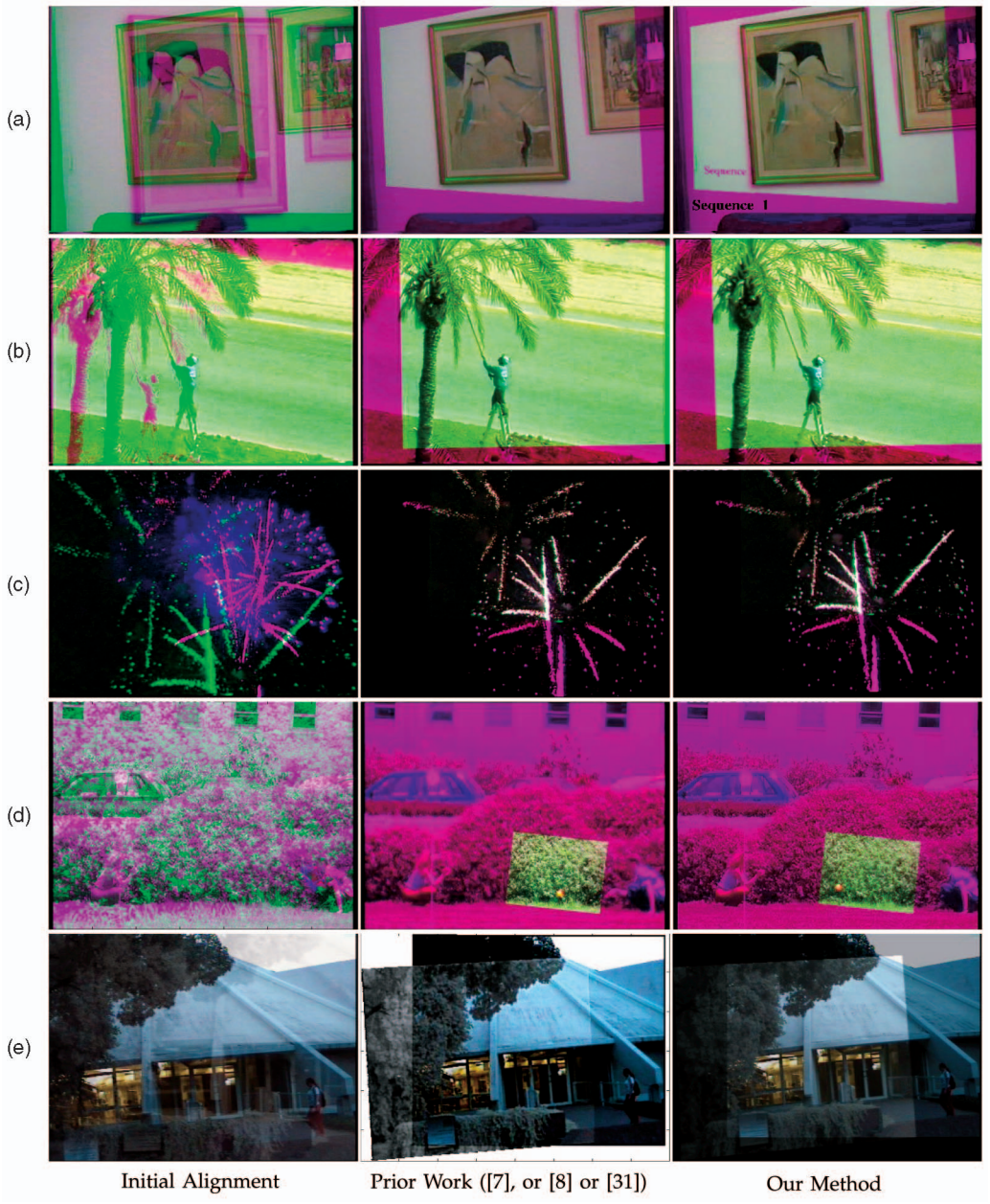


Fig. 9. Comparison with [7]: (a) light, (b) palm, and (c) firework sequences. Comparison with [8] (d) wide sequence. Comparison with [31]: (e) library sequence.

TABLE 4
Percentage Inliers from Dynamic Appearance for Different Sequences

Type	Rigid			Non-Rigid			Special	
Sequence	Palm	Parking	Light	Fountain	Fireworks	Flag	Wide	Multimodal
Inlier Percentage	68.75	78.74	84.66	93.80	93.90	100.00	71.02	89.90

method reduces the number of frames we need to process. In the case of [7], one needs to perform feature extraction, tracking, and trajectory matching for two sets of F frames. In our case, we only need feature extraction over two sets of $n+1 \ll F$ images. In addition, we have the computations for calculating the system parameters. Typically, F/n for the sequences we have presented is in the range of 8-10; hence, using our method gives an advantage with respect to the number of frames we need to process. We do not need to make the choice of whether to register using only the mean image or only the C matrix of the LDS or by using both. Given the information extracted from the video sequences, the algorithm automatically makes this choice. This gives us a generic algorithm that can be applied to both rigid and nonrigid sequences. In short, we have presented a method that works equally well compared to the state of the art but is more efficient.

The other two important contributions of this paper are the use of a joint system identification framework together with a canonical form representation. The joint identification

and the Jordan canonical form are not only applicable to the case of registering video sequences, but also to the entire genre of algorithms based on the dynamic texture model. In this paper, we have also shown that out of all the possible choices for the method of identification and canonical form, the JID using JCF performs the best.

ACKNOWLEDGMENTS

This work was partially supported by startup funds from the Johns Hopkins Univerity, by grants US Office of Naval Research (ONR) N00014-05-10836, the US National Science Foundation CAREER 0447739, and ONR N00014-09-1-0084.

REFERENCES

[1] http://www.wisdom.weizmann.ac.il/vision/VideoAnalysis/Demos/Seq2S eq/, 2008.
[2] http://www.wisdom.weizmann.ac.il/vision/VideoAnalysis/Demos/Traj2Traj, 2010.
[3] http://www.wisdom.weizmann.ac.il/vision/SpaceTimeCorrelations.html, 2010.
[4] A. Agarwala, K.C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic Video Textures," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 821-827, 2005.
[5] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture Mixing and Texture Movie Synthesis Using Statistical Learning," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 2, pp. 120-135, Apr.-June 2001.
[6] M. Brown, R. Szeliski, and S. Winder, "Multi-Image Matching Using Multi-Scale Oriented Patches," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 510-517, June 2005.
[7] Y. Caspi and M. Irani, "Spatio-Temporal Alignment of Sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1409-1424, Nov. 2002.
[8] Y. Caspi, D. Simakov, and M. Irani, "Feature-Based Sequence-to-Sequence Matching," *Int'l J. Computer Vision,* vol. 68, no. 1, pp. 53-64, 2006.
[9] A. Chan and N. Vasconcelos, "Classifying Video with Kernel Dynamic Textures," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-6, 2007.
[10] A. Chan and N. Vasconcelos, "Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 846-851, 2005.
[11] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic Textures," *Int'l J. Computer Vision,* vol. 51, no. 2, pp. 91-109, 2003.
[12] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic Texture Segmentation," *Proc. IEEE Conf. Computer Vision,* pp. 44-49, 2003.
[13] G. Doretto and S. Soatto, "Dynamic Shape and Appearance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2006-2019, Dec. 2006.
[14] M.A. Fischler and R.C. Bolles, "RANSAC Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 26, pp. 381-395, 1981.
[15] A. Fitzgibbon, "Stochastic Rigidity: Image Registration for Nowhere-Static Scenes," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 662-669, 2001.
[16] C. Harris and M. Stephens, "A Combined Corner and Edge Detection," *Proc. Fourth Alvey Vision Conf.,* 1988.
[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge Univ. Press, 2000.
[18] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics,* vol. 22, pp. 277-286, 2003.
[19] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 20, pp. 91-110, 2003.
[20] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models.* Springer Verlag, 2003.
[21] P.V. Overschee and B.D. Moor, "Subspace Algorithms for the Stochastic Identification Problem," *Automatica,* vol. 29, no. 3, pp. 649-660, 1993.
[22] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Dynamosaics: Video Mosaics with Non-Chronological Time," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 58-65, 2005.
[23] A. Rav-Acha, Y. Pritch, and S. Peleg, "Online Registration of Dynamic Scenes Using Video Extrapolation," *Proc. Workshop Dynamic Vision at IEEE Int'l Conf. Computer Vision,* 2005.
[24] A. Ravichandran and R. Vidal, "Mosaicing Nonrigid Dynamical Scenes," *Proc. Workshop Dynamic Vision,* 2007.
[25] W.J. Rugh, *Linear System Theory,* second ed. Prentice Hall, 1996.
[26] P. Saisan, G. Doretto, Y.N. Wu, and S. Soatto, "Dynamic Texture Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 58-63, 2001.
[27] A. Schödl, R. Szeliski, D.H. Salesin, and I. Essa, "Video Textures," *Proc. ACM SIGGRAPH,* pp. 489-498, 2000.
[28] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1994.
[29] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Fundamental Trends in Computer Graphics and Vision,* vol. 2, no. 1, pp. 1-104, 2006.
[30] M. Szummer and R.W. Picard, "Temporal Texture Modeling," *Proc. IEEE Int'l Conf. Image Processing,* vol. 3, pp. 823-826, 1996.
[31] Y. Ukrainitz and M. Irani, "Aligning Sequences and Actions by Maximizing Space-Time Correlations," *Proc. European Conf. Computer Vision,* pp. 538-550, 2006.
[32] R. Vidal and P. Favaro, "Dynamicboost: Boosting Time Series Generated by Dynamical Systems," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.
[33] R. Vidal and A. Ravichandran, "Optical Flow Estimation and Segmentation of Multiple Moving Dynamic Textures," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 516-521, 2005.
[34] P.A. Viola, "Alignment by Maximization of Mutual Information," Technical Report AITR-1548, 1995.
[35] L. Wei and M. Levoy, "Fast Texture Synthesis Using Tree-Structured Vector Quantization," *Proc. ACM SIGGRAPH,* pp. 479-488, 2000.

Avinash Ravichandran received the BE degree in electronics and communication engineering from the University of Madras in 2003, and the master's degree in electrical engineering in 2007 and the master's degree in applied mathematics and statistics in 2009 from The Johns Hopkins University, where he is currently working toward the PhD degree in the Vision, Learning, and Dynamics Lab. His research interests include dynamic texture classification, registration, and segmentation. He is a student member of the IEEE.

René Vidal received the BS degree in electrical engineering (highest honors) from the Pontificia Universidad Catolica de Chile in 1997, and the MS and PhD degrees in electrical engineering and computer sciences from the University of California at Berkeley in 2000 and 2003, respectively. He was a research fellow at the National ICT Australia in the Fall of 2003 and joined The Johns Hopkins University in January 2004 as an assistant professor in the Department of Biomedical Engineering and the Center for Imaging Science. He has coauthored more than 100 articles in biomedical image analysis, computer vision, machine learning, hybrid systems, and robotics. He is the recipient of the 2009 US Office of Naval Research Young Investigator Award, the 2009 Sloan Research Fellowship, the 2005 US National Science Foundation (NFS) CAREER Award, and the 2004 Best Paper Award Honorable Mention (with Professor Yi Ma) for his work on "A Unified Algebraic Approach to 2D and 3D Motion Segmentation" presented at the European Conference on Computer Vision. He also received the 2004 Sakrison Memorial Prize for "completing an exceptionally documented piece of research," the 2003 Eli Jury award for "outstanding achievement in the area of Systems, Communications, Control, or Signal Processing," the 2002 Student Continuation Award from NASA Ames, the 1998 Marcos Orrego Puelma Award from the Institute of Engineers of Chile, and the 1997 Award of the School of Engineering of the Pontificia Universidad Catolica de Chile to the best graduating student of the school. He is a member of the IEEE and the ACM.