# Video Registration using Dynamic Textures

Avinash Ravichandran and René Vidal

Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218, USA

**Abstract.** We propose a dynamic texture feature-based algorithm for registering two video sequences of a rigid or nonrigid scene taken from two synchronous or asynchronous cameras. We model each video sequence as the output of a linear dynamical system, and transform the task of registering frames of the two sequences to that of registering the parameters of the corresponding models. This allows us to perform registration using the more classical image-based features as opposed to space-time features, such as space-time volumes or feature trajectories. As the model parameters are not uniquely defined, we propose a generic method to resolve these ambiguities by jointly identifying the parameters from multiple video sequences. We finally 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.

## 1 Introduction

A classical problem in computer vision is aligning two images of the same scene, taken from different viewpoints. This problem is known as image registration, and its objective is to recover the spatial alignment between the two images.

There exists a vast amount of literature on the problem of image registration for both rigid and nonrigid scenes (see [1] and the references therein). For example, feature-based approaches to image registration proceed by first extracting features such as Harris Corners [3], SIFT [2], MOPS [4], etc., from the images. The features from the first image are then matched to the features from the second image. The alignment between the two images is recovered from these matches using methods such as RANSAC [5]. A homography or an affine transformation is then computed by fitting such a model to the inliers given by RANSAC.

The last few years have seen an increasing interest in the more challenging problem of registering video sequences. In addition to recovering the spatial transformation, video registration also involves recovering the temporal alignment between the two videos. Although this poses additional challenges with respect to image registration, one can still argue that image registration algorithms can be extended in a straightforward manner to solve the video registration problem. When there is no temporal lag/shift between the two video sequences, the problem reduces to aligning corresponding pairs of images from the two video sequences. That is, one can extract features from every frame of the two video sequences, match the features from a frame in the first sequence to the features from the corresponding frame in the second sequence, and then recover a spatial transformation by applying RANSAC to all features from all pairs of frames. When there is a temporal lag between the two videos, one could follow the same procedure. However, the correspondence between frames of the two video sequences is unknown. This correspondence can be found using all possible choices of frame pairs from the two video sequences. This exhaustive search approach will indeed lead to the optimal result. However, this comes at the price of the computational cost involved in solving this brute force optimization.

To circumvent this problem, Caspi et al. [6, 7] use feature trajectories rather than features from every frame. Feature points are extracted from the first frame and tracked throughout the video. The registration problem is posed as a trajectory matching problem between the two video sequences. An optimization problem is then solved to jointly recover the spatial and temporal transformation between the video sequences. In some sense the tracking algorithm performs the matching between two consecutive frames, which is an easier problem compared to matching features across non corresponding frames. Ukrainitz et al. [8] on the other hand, work with dense space-time information instead of feature trajectories. The idea here is to maximize a global similarity measure between small space-time volumes extracted from the two videos using a costly optimization scheme.

**Paper contributions.** In this paper, we propose a much simpler video registration framework, which requires neither feature tracking nor a costly optimization procedure. Instead, we reduce the spatial-temporal video registration problem to the spatial registration of a collection of image bases extracted from the video.

Similar to [9], we model the two video sequences as the output of two linear dynamical systems (LDSs). Since the two cameras observe the same dynamical scene, but from a different viewpoint, we assume that the dynamics of the image intensities in the two views are the same. Our first contribution is to propose a simple scheme for the joint identification of the two LDSs. The proposed method retains the sub-optimality of the original identification algorithm of Doretto et al. [10], but at the same time enforces the criterion that the dynamics of the two video sequences must be the same.

Since the parameters of the identified LDSs do not depend on the temporal alignment, the spatial registration of the two videos can be obtained by applying classical image registration techniques to the parameters of the LDSs. However, since the parameters of the LDSs are computed only up to change of basis, they need to be transformed to a canonical form before they can be compared. Our second contribution is to propose the use of the Jordan canonical form (JCF), together with a method for converting the identified parameters to the JCF. This converts all the parameters into the same basis and makes comparing the parameters more straightforward. The method is numerically stable, independent of a reference sequence and scales very well for an arbitrary number of sequences.

We finally propose a feature-based method for registering multiple video sequences using the dynamic texture model. Our approach does not work with space-time volumes [7] or feature trajectories [8], and does not rely on heuristics such as the appearance image [9]. Instead, we extract standard image based features from the parameters of the LDSs to perform the registration.

## 2 Video Registration using Dynamic Textures

### 2.1 Dynamic Textures Framework

We first briefly review the dynamic texture model. Please see [10] for more details. Given a video sequence $\{I(t) \in \mathbb{R}^p\}_{t=1}^F$, we model it as the output of a LDS

$$z(t+1) = Az(t) + Bv(t) \tag{1}$$

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

The vector $z(t) \in \mathbb{R}^n$ 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* $B \in \mathbb{R}^{n\times q}$. The zero-mean Gaussian processes $v(t) \sim \mathcal{N}(0,Q)$ and $w(t) \sim \mathcal{N}(0,R)$ model the process noise and the measurement noise, respectively. The *appearance matrix* $C \in \mathbb{R}^{p\times n}$ maps the hidden state to the image and the vector $C^0 \in \mathbb{R}^p$ is the temporal mean of the video sequence. 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 of the video, represented by $C$, from the temporal evolution of the video represented by $A$. This property will allow us to recover the spatial registration independently from the temporal lag between the video sequences.

Given $\{I(t)\}_{t=1}^F$, 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 [11]. 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 [10]. 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 $C^0$ is first calculated. The parameters of the system are then identified using SVD to factorize the mean subtracted matrix as

$$\left[I(1) - C^0, \ldots, I(F) - C^0\right] = U(SV^\top) = CZ. \tag{3}$$

Given $Z = [z(1) \ldots z(F)]$, the parameter $A$ is obtained as the least squares solution to the system of linear equations $A[z(1) \ldots z(F-1)] = [z(2) \ldots z(F)]$.

It is well known that if one performs the SVD, the factorization is unique up to an invertible transformation, i.e. the recovered factors are $CP^{-1}$ and $PZ$, 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, if 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. This poses a problem when one wants to compare the parameters say column by column. In order to address this issue, in the next section we outline a method to account for the change of basis. We propose to do this by converting the parameters into a canonical form. Once the parameters are in the canonical form, the task of comparing these parameters becomes straightforward.

### 2.2 Jordan Canonical Form for Parameter Comparison

There have been a few attempts to address the basis issue in prior work. In [12], Chan et al. used one sequence as the reference and converted the other sequence into the basis of the reference sequence. Vidal et al. in [13] used the diagonal form of the $A$ matrix. The problem is that the resulting parameters in canonical form are complex, because the eigenvalues of $A$ can be complex. To deal with this issue, the Reachability Canonical Form (RCF) was used in [9]. However, the RCF form uses the pair $(A, B)$ to convert the system into canonical form. For 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 $B$, which models the input noise, is not that critical to describe the appearance of the scene. An obvious alternative is to use the Observability Canonical Form (OCF) [14]

$$A_c = \begin{bmatrix} -a_{n-1} & 1 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1 & 0 & 0 & \vdots & 1 \\ -a_0 & 0 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n} \quad \text{and} \quad C_c = \begin{bmatrix} 1 \; 0 \; 0 \cdots 0 \end{bmatrix} \in \mathbb{R}^{1\times n}, \tag{4}$$

where $A^n + a_{n-1}A^{n-1} + \ldots + a_0 I = 0$ is the characteristic polynomial of $A$. However, it is well known that using the OCF can lead to numerical instabilities [14].

In order to overcome this issue, we propose to use a canonical form based on the Jordan real form. When $A$ has $2p$ complex eigenvalues and $n - 2p$ real eigenvalues, the Jordan Canonical Form (JCF) is given by

$$A_c = \begin{bmatrix} \sigma_1 & \omega_1 & 0 & \cdots & 0 \\ -\omega_1 & \sigma_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \lambda_{2p-n-1} & 0 \\ 0 & 0 & \dots & 0 & \lambda_{2p-n} \end{bmatrix} \quad \text{and} \quad C_c = \begin{bmatrix} 1 \; 0 \; 1 \; 0 \dots 1 \end{bmatrix}, \tag{5}$$

where the eigenvalues of $A$ are given by $\{\sigma_1 \pm i\omega_1, \ldots, \sigma_p \pm i\omega_p, \lambda_1, \ldots, \lambda_{n-2p}\}$. Note that the JCF is equivalent to the RCF or the OCF, but in a different basis.

Given any canonical form, we now outline the steps that we need to take in order to convert the identified parameters into the canonical form. Assume that we have the identified parameters $(A, C)$. We 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 it to be $\begin{bmatrix} 1 \; 1 \dots 1 \end{bmatrix}^\top$, so that all rows of $C$ are weighted equally. Now, the relationship between the $A$ matrix and its canonical form $A_c$ is a special form of the Sylvester equation

$$A_c P - PA = 0. \tag{6}$$

Vectorizing this equation, we obtain $(I \otimes A_c - A^\top \otimes I)\text{vec}(P) = 0$, where $\otimes$ represents the Kronecker product. If we also vectorize the equation relating the $C$ matrices, $C_c P = \gamma^\top C$, we can solve for $P$ from both equations as follows

$$\begin{bmatrix} I \otimes A_c - A^\top \otimes I \\ I \otimes C_c \end{bmatrix} \text{vec}(P) = \begin{bmatrix} 0 \\ C^\top \gamma \end{bmatrix}. \tag{7}$$

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 the eigenvalues are different, we can choose a predefined way to sort the eigenvalues, and obtain a unique JCF.

### 2.3 Joint Identification of Dynamic Textures

In this section, we will show that using a canonical form helps us to jointly identify the parameters of the LDSs associated with multiple video sequences of the same rigid or nonrigid scene, but taken by multiple (asynchronous) cameras. We will model such a collection of $m$ video sequences $\{I_i(t) \in \mathbb{R}^{p_i}\}_{t=1,\ldots,F}^{i=1,\ldots m}$ as the output of $m$ LDSs whose dynamics are the same. Specifically, the LDSs share the same $A$ matrix and the same state $z(t)$, modulo a temporal shift $\tau_i \in \mathbb{Z}$, i.e.

$$z(t+1) = Az(t) + Bv(t) \quad \text{and} \quad I_i(t) = C_i^0 + C_i z(t + \tau_i) + w_i(t). \tag{8}$$

One possible way of identifying the parameters these $m$ LDSs is to apply the identification algorithm described in §2.1 to each video sequence separately. More specifically, we let $\tilde{I}_i(t) = I_i(t) - C_i^0$, form the matrix $W_i = [\tilde{I}_i(1) \ldots \tilde{I}_i(F)]$, and calculate its SVD $W_i = U_i S_i V_i^\top$. The parameters of the LDSs are identified as $C_i = U(:, 1:n)$ and $Z_i = S(1:n, 1:n)V(:, 1:n)^\top$. The matrix $A_i$ is then linearly obtained from $Z_i$. An immediate problem with this scheme is that, due to the presence of noise, viewpoint changes and the suboptimal identification, there is no guarantee that the $A_i$ matrices for the $m$ videos will be the same.

In order to enforce the same dynamics, we propose a simple method where all the videos are stacked into a single matrix $W$, which is factorized using the SVD as

$$W = \begin{bmatrix} \tilde{I}_1(1) \ \ldots \ \tilde{I}_1(F) \\ \vdots \\ \tilde{I}_m(1) \ldots \tilde{I}_m(F) \end{bmatrix} = USV^\top. \tag{9}$$

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 $z(t) = A^t z_0$, where $z_0$ is the initial state of the system. Now if we consider a temporal lag $\tau_i \in \mathbb{Z}$ for the $i^{th}$ video sequence, then the evolution of the hidden state of the $i^{th}$ sequence is given by $z_i(t) = A^{\tau_i} z(t)$. Therefore, we can decompose $W$ as follows

$$W = \begin{bmatrix} C_1 A^{\tau_1} z(1) \ \ldots \ C_1 A^{\tau_1} z(F) \\ C_2 A^{\tau_2} z(1) \ \ldots \ C_2 A^{\tau_2} z(F) \\ \vdots \\ C_m A^{\tau_m} z(1) \ldots C_m A^{\tau_m} z(F) \end{bmatrix} = \begin{bmatrix} C_1 A^{\tau_1} \\ C_2 A^{\tau_2} \\ \vdots \\ C_m A^{\tau_m} \end{bmatrix} \begin{bmatrix} z(1) \ z(2) \ldots z(F) \end{bmatrix}. \tag{10}$$

Note that the $C_i$ matrix we recover from stacking all the videos is the true $C_i$ matrix multiplied by $A^{\tau_i}$. The problem is that $A$ is unknown, and so we cannot directly compute $C_i$. To resolve this, consider the equation for the $i^{th}$ video

$$[\tilde{I}_i(1)\ldots\tilde{I}_i(F)]=C_iA^{\tau_i}[z(1)\ldots z(F)]=C_iA^{\tau_i}(A^{\tau_i})^{-1}[z(\tau_i+1)\ldots z(F+\tau_i)]. \quad (11)$$

We can 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 $A^{\tau_i}$ and recover the original parameters in their canonical form. The details can be found in Algorithm 1. Having identified a dynamic texture model for all video sequences with a common $A$, and all the $C$ matrices with respect to the same basis, in the next section we describe a method to register multiple video sequences using the joint identification framework.

### 2.4 Registering Video Sequences using Dynamic Textures

Consider two video sequences $I_1(x,t)$ and $I_2(x,t)$, where $x$ denotes the pixel coordinates, and $t=1,\ldots,F$. Assume that the video sequences are related by a spatial transformation $H$ and a temporal lag $\tau$, i.e. $I_1(x,t)=I_2(H(x),t+\tau)$. It follows from (8) that $\tau=\tau_1-\tau_2$, $C_1^0(x)=C_2^0(H(x))$, and $C_1(x)=C_2(H(x))$. Therefore, the joint identification algorithm described in §2.3 allows us to decouple the recovery of the spatial alignment from the temporal lag. Based on this fact, we now propose a straightforward method to spatially register two video sequences using both the *mean image* $C^0$ and the $n$ columns of the $C$ matrix, which we will refer to as the *dynamic appearance images*. Once the spatial alignment has been recovered, we temporally align the two sequences using a simple line search in the temporal direction, i.e. $\tau=\arg\min_\tau\sum_t\|I_1(\mathbf{x},t)-I_2(H(x),t+\tau)\|^2$.

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 $i^{th}$ column of the $j^{th}$ sequence represented as an image. We use a feature-based approach to spatially register the two set 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 $i\in\{0,\ldots,n\}$. 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 co-ordinates and $M$ is the total matches from the $n+1$ image pairs. We then 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 non linear method outlined in [15]. Notice that, by following this approach, we weight the contribution of the correspondences from every image pair equally. This is because 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 [9]. This choice now becomes automatic.

**Algorithm 1**: **(Joint identification of multiple video sequences)**

```
Given $I_i(t)$, first calculate $C_i^0$, the temporal image mean. Set $\tilde{I}_i(t) = I_i(t) - C_i^0$.
Compute $C, Z$ using the rank $n$ singular value decomposition of the matrix
    $W = \begin{bmatrix} \tilde{I}_1(1), \ \dots \ , \tilde{I}_1(F) \\ \vdots \\ \tilde{I}_m(1), \dots , \tilde{I}_m(F) \end{bmatrix} = USV^\top \quad Z = SV^\top, \quad C = U.$
Compute $A = [z(2), \dots, z(F)][z(1), \dots, z(F-1)]^\dagger$.
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 canonical form.
```

**Algorithm 2**: **(Registration of multiple video sequences)**

```
Given $I_1(t)$ and $I_2(t)$, calculate the parameters $A$ and $(C_i^0, C_i)$.
Extract features and feature descriptors from $(C_j^i), j = \{1, 2\}, i = 0, \dots, n$.
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 it's corresponding match into $X_2$
Recover the Homography $H$ using RANSAC such that $X_2 \sim HX_1$.
Calculate temporal alignment $\tau$ as $\tau = \arg\min_\tau \sum_t \|I_1(\mathbf{x}, t) - I_2(H(x), t + \tau)\|^2$.
```

## 3 Experimental Results

In this section, we evaluate the different steps of our algorithm on a wide variety of sequences and compare its performance to existing algorithms.

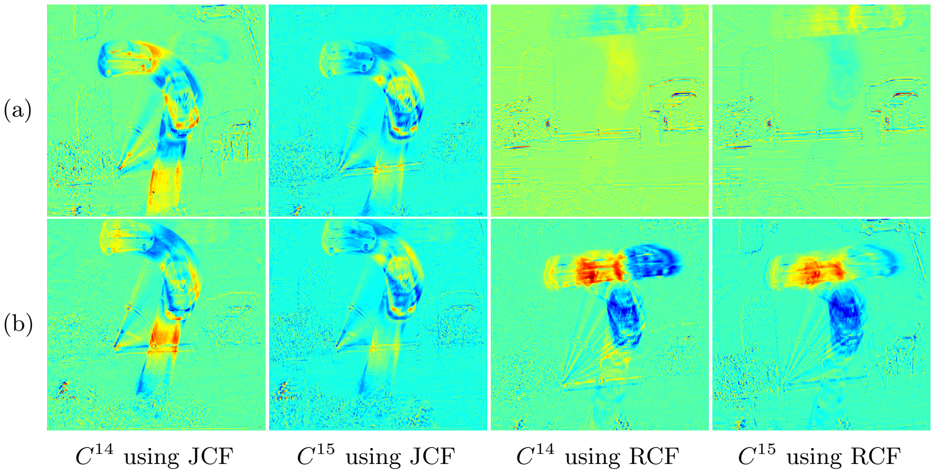
**Evaluation of the Jordan Canonical Form.** To evaluate our canonical form, we first take a video sequence and identify the parameters of the system using the suboptimal approach. We apply different kinds of transformations to the parameters of the system, i.e. given $(C, A)$ we form new parameters $(\tilde{C}, \tilde{A}) = (CP^{-1}, PAP^{-1})$. This simulates the ambiguities we encounter using the suboptimal approach. 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 are then calculated between the parameters before and after converting it to the canonical form. We perform the experiment for 200 trials of each transformation and then calculate the mean error. The transformations are randomly generated from three different classes of transformations: an invertible matrix, an orthogonal matrix or a sign flip. 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 simple transformations such as the sign flip, both the canonical forms perform very well. However, when the transformation gets more involved, 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 show two sample columns from the $C$ matrix of two video sequences from the

**Table 1.** Parameter errors before and after converting to the canonical form.

| Transformation | Errors initially | | Errors in JCF | | Errors in RCF | |
|---|---|---|---|---|---|---|
| | $\|A-\tilde{A}\|_F$ | $\|C-\tilde{C}\|_F$ | $\|A-\tilde{A}\|_F$ | $\|C-\tilde{C}\|_F$ | $\|A-\tilde{A}\|_F$ | $\|C-\tilde{C}\|_F$ |
| Sign flip | 2.568 | 7.063 | 0 | 0 | 0 | 0 |
| Invertible | 1.16e+02 | 1.51e+02 | 6.79e-10 | 1.31e-04 | 2.00e+06 | 1.08e+08 |
| Orthogonal | 2.61e+00 | 7.06e+00 | 8.04e-14 | 1.31e-08 | 1.48e+07 | 6.46e+08 |

parking set, which we will later show registration results on. Fig. 1 shows columns of $C$ identified using both algorithms. One can see that the corresponding column images are spatially transformed versions of each other when we use the JCF. However, no such trend is observed when we use the RCF.



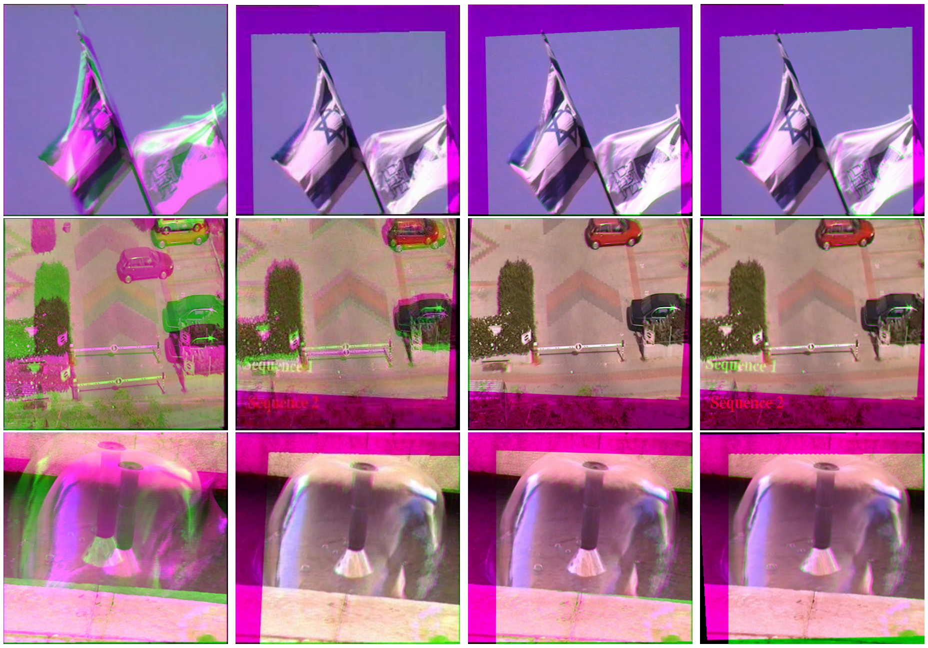
$C^{14}$ using JCF $C^{15}$ using JCF $C^{14}$ using RCF $C^{15}$ using RCF

**Fig. 1.** Sample columns of the $C$ matrix displayed as images (a) Sequence 1 from the parking set (b) Sequence 2 from the parking set.

**Evaluation of the Registration.** For our experiments on registration, we tested our algorithm on all the sequences available at [16], one sequence from [17] and one from [18]. These test video sequences consist of 3 pairs of sequences of rigid scenes, 3 pairs of sequences of nonrigid scenes, and 2 other pairs of sequences. Our last category had one pair of sequences with a relatively large zoom factor and the other was from two non stationary cameras capturing different modalities.

We use the *composite video* in order to qualitatively asses the performance of the registration. The composite video is formed by taking the red and blue color channels from the first video and the green color channel from the second video. We show the composite video before and after registration so as to display both the initial and final alignments. Notice that green and purple regions in the composite video correspond to errors in the registration, or regions that exist in one image, but not in the other.

We first compare the methods of [9] and [6] to ours on the common sequences from [9] and [6], i.e. the flag, parking, and fountain sequences. We see from Fig. 2 that for all the 3 sequences the alignment we obtain is as good if not better that that obtained by the other methods. Moreover, when compared to [9], we are able to perform the registration with lower model orders for the LDSs. The model order for [9] was in the range 50-75, but in our method they were in the range of 20-30. In comparision with [6], our algorithm avoids expensive optimizations and is able to recover the same quality of alignment from much fewer images.



Initial Alignment | Ravichandran [9] | Caspi [6] | Our Method

**Fig. 2.** Comparison of results from three methods: First Row: Flag sequence, Second Row: Parking sequence and Last Row: Fountain sequence.

We now show additional results in Fig. 3. The sequences here exhibit different kinds of variations such as, variation in intensity, shape (non-rigid objects) and modality. We see that in all the cases we perform as well as state-of-the-art methods. We are able to register images with a large baseline transformation as shown in Fig. 3(d). Although we have not taken any explicit measure to account for multi-modality, we found that by using SIFT features we are able to register multimodal video sequences as shown in Fig. 3(e). Here, one sequence is captured using a normal camera and the other is captured using an infrared camera. For this case, one can notice that our result and the result from [8] do not look the same. This is because we do not perform any kind of fusion.

In order to further analyze our registration algorithm, we obtained the inliers from RANSAC and then calculated the percentage of inliers from the mean image and from the dynamic appearance images. This result can be seen in Table 2. The interesting fact about these numbers is that our algorithm automatically adapts the percentage of features obtained from the mean and dynamic appearance images. Notice that the percentage of inliers from the dynamic appearance images is 100% for the flag sequence. This is in agreement from the conclusions of [9], where the authors reported that by using the appearance image alone they were able to register the sequences the best. Thus we see that the algorithm performs very well on a large variety of sequences and our results are comparable to existing methods.

**Table 2.** Percentage inliers from dynamic appearance images for different sequences.

| Type | Rigid | | | Non-Rigid | | | Other | |
|---|---|---|---|---|---|---|---|---|
| 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 |

## 4 Discussion and Conclusion

We have proposed a method for registering video sequences based on the dynamic texture model. As compared to [6], we see that 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 method reduces the number of frames we need to process. In the case of [6], they need 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 number of frames we need to process. As compared to [9], we do not need to make the choice of whether to register using the mean or the dynamic appearance or 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 non-rigid sequences. One other advantage is that we are able to register the video sequences with lower system orders. In short, we have presented an efficient method that works equally well compared to the state of the art.

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 canonical form are not only applicable to the case of registering video sequences, but also to the entire genre of algorithms

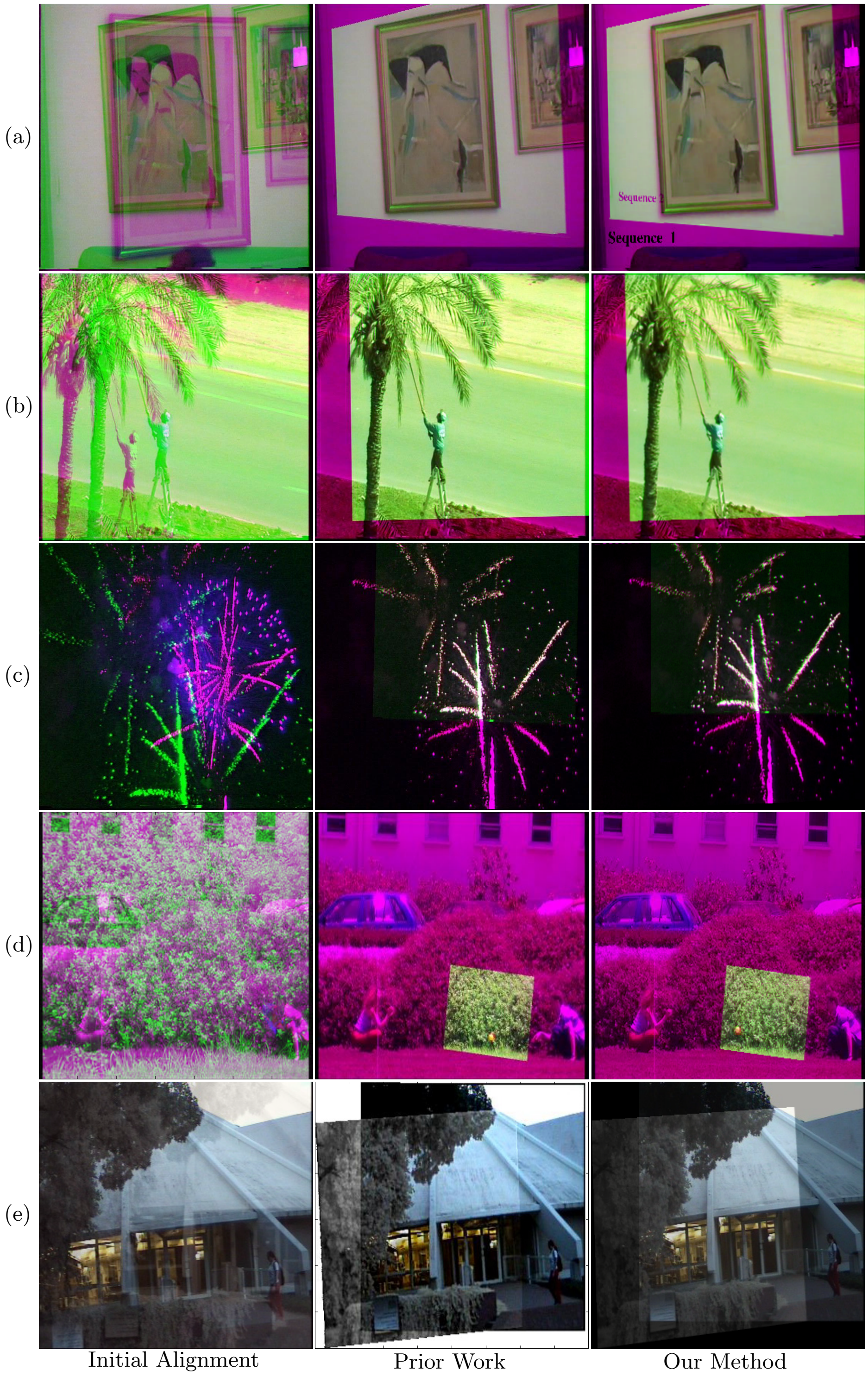


**Fig. 3.** Comparison with [6] (a) Light (b) Palm (c) Firework Sequences. Comparison with [7] (d) Wide Zoom Sequence. Comparison with [8] (e) Library Sequence

based on the dynamic texture model. Our future work involves investigating the performance gain obtained by this framework on other problems such as dynamic texture recognition.

## Acknowledgments

This work was partially supported by startup funds from JHU, by grants ONR N00014-05-1083, NSF CAREER 0447739, NSF EHS-0509101, and by contract JHU APL-934652.

## References

1. Szeliski, R.: Image alignment and stitching: A tutorial. Fundamental Trends in Computer Graphics and Vision **2** (2006) 1–104
2. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision. Volume 20. (2003) 91–110
3. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of The Fourth Alvey Vision Conference. (1988)
4. Brown, M., Szeliski, R., Winder, S.: Multi-image matching using multi-scale oriented patches. In: CVPR. (2005) 510–517
5. Fischler, M.A., Bolles, R.C.: RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **26** (1981) 381–395
6. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1409–1424
7. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. International Journal of Computer Vision **68** (2006) 53–64
8. Ukrainitz, Y., Irani, M.: Aligning sequences and actions by maximizing space-time correlations. In: European Conference on Computer Vision. (2006) 538–550
9. Ravichandran, A., Vidal., R.: Mosaicing nonrigid dynamical scenes. In: Workshop on Dynamic Vision. (2007)
10. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. International Journal of Computer Vision **51** (2003) 91–109
11. Overschee, P.V., Moor, B.D.: N4SID : Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, Special Issue in Statistical Signal Processing and Control (1994) 75–93
12. Chan, A., Vasconcelos, N.: Probabilistic kernels for the classification of autoregressive visual processes. In: Conference on Computer Vision and Pattern Recognition. Volume 1. (2005) 846–851
13. Vidal, R., Ravichandran, A.: Optical flow estimation and segmentation of multiple moving dynamic textures. In: Conference on Computer Vision and Pattern Recognition. Volume II. (2005) 516–521
14. Rugh, W.J.: Linear System Theory. Second edn. Prentice Hall (1996)
15. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge (2000)
16. http://www.wisdom.weizmann.ac.il/∼vision/VideoAnalysis/Demos/Seq2Seq/.
17. http://www.wisdom.weizmann.ac.il/∼vision/VideoAnalysis/Demos/Traj2Traj.
18. http://www.wisdom.weizmann.ac.il/∼vision/SpaceTimeCorrelations.html.