Mosaicing Non-Rigid Dynamical Scenes

Avinash Ravichandran and René Vidal

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

Abstract. In this paper, we deal with the problem of spatially and temporally registering multiple video sequences of a non-rigid dynamical scene. For example, registering multiple videos of a fountain taken from different vantage points. Our approach is not based on frame-by-frame or volume-by-volume registration. Instead, we use the dynamic texture framework, which models the non-rigidity of the scene with linear dynamical systems encoding both the dynamics and the appearance of the scene. Our key contribution is to observe that a certain appearance matrix extracted from the dynamic texture model is invariant with respect to the non-rigid motions of the scene, thus it can be directly used to register the video sequences. Our framework is applicable to both synchronized videos as well as videos containing temporal lags. In the latter case, we also propose a method to synthesize novel sequences without the temporal lags. We then show how our model can be extended to the case where there is camera motion in the video sequences. The final result is a simple and flexible method that achieves state-of-the-art performance with a significant reduction in computational complexity.

1 Introduction

Image registration refers to the problem of finding correspondences between two or more images. Once such correspondences have been found, all images can be put into the same reference frame. Image registration finds a wide variety of applications in computer vision, especially in mosaicing, image-based modeling and rendering, structure-from-motion, object recognition, etc. Image registration is also important in medical imaging, where multi-modal 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 [10], SIFT features [13], etc., are first extracted from both images. These features are then matched using methods such as normalized cross correlation. Once a rough (possibly incorrect) matching is obtained, one can use methods such as RANSAC [8] to refine the matching and calculate the transformation between the two images. Direct methods on the other hand, first define a metric, such as the sum of square differences, mutual information [22], etc. The registration problem is then solved by minimization/maximization of a cost function built from this metric.

In this paper we address the more general problem of registering two video sequences of a non-rigid dynamical scene. This poses two main challenges with respect to classical image registration methods. First, we need to find the temporal alignment in addition to the spatial alignment. Second, we are dealing with non-rigid scenes. Therefore, we cannot register them using classical constraints such as the brightness constancy constraint, which are applicable only in the rigid case.

If the two sequences were temporally aligned, one could argue that the video registration problem could be reduced to an image registration problem, especially when the scene is predominantly rigid. However, the main challenge is to decide which pair of frames should be registered. While in principle any pair would do, there is no reason to believe that registering the first frame from each sequences will give the optimal solution. 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. Now, if the dynamics of the scene need to be taken into account, the video sequence can be considered as a space-time volume. Then volume-to-volume registration can be performed using either the entire volume or a collection of sub-volumes, or point trajectories (see §1.2). However, such methods suffer from the same drawbacks. First, they are computationally intensive as they involve an optimal search over the entire video. Second, it is hard to define a good matching criteria among sub-volumes, because the brightness constancy constraint does not hold.

In some sense, registration of dynamical scenes is a chicken-and-egg problem: if one is given the spatial alignment one can easily calculate the temporal alignment and vice versa. Since recovering the spatial alignment is the harder of the two problems, in this paper we will focus on the former problem. However, we do not assume that the temporal alignment is known. Instead, we approach the problem in a such a way that recovering the spatial alignment is independent of the temporal alignment.

1.1 Paper Contributions

In this paper we propose a method for registering non-rigid dynamical scenes that is computationally simple and does not rely on frame-to-frame or volume-to-volume matching. We solve this problem by taking a modeling approach, which effectively reduces the complicated problem of spatial-temporal alignment of video sequences to the well solved problem of spatially aligning images. More specifically, we model the temporal evolution of a non-rigid dynamical scene using a linear dynamical system (LDS). The LDS captures both the non-rigid dynamics of the scene as well as its appearance. Since the appearance does not depend on the non-rigid motions, one can align the two videos by registering a pair of images built from the appearance matrices of the videos. As a consequence, the method is applicable to both synchronized and unsynchronized cameras, because the appearance matrices are invariant to temporal lags, and one can recover the spatial transformation without recovering the temporal lag. In particular, the proposed approach is applicable in the following scenarios.

a. **Multiple synchronized cameras capturing non-rigid scenes:** Here we have multiple static and temporally synchronized cameras capturing video sequences of a non-rigid dynamical scene For instance, two synchronized cameras viewing a river.

The only motion in these sequences is due to the non-rigidity of the objects in the scene. Although this is a simple case, we use this case to motivate our framework.

b. **Multiple unsynchronized cameras capturing non-rigid scenes:** Here we have multiple static cameras capturing a non-rigid dynamical scene. However, the cameras are not synchronized, e.g., a couple recording a firework show. This case is also equivalent to having a single camera capturing the scene at different time intervals.
c. **Single moving camera capturing a non-rigid scene:** Here we have a single camera panning over a non-rigid scene such as the ocean or a bed of flowers etc. We then want to generate a mosaic from the video sequence of these non-rigid scenes.

1.2 Related Work

Over the past few years, several methods for modeling non-rigid dynamical scenes have been proposed [19, 17, 23, 2, 5, 11]. All 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 [5] is particularly attractive, because the parameters of the model can be clearly interpreted for the purposes of synthesis, segmentation, classification and recognition.

One of the first methods for registering non-rigid dynamical scenes was proposed by Fitzgibbon [9]. The method combines dynamic textures with stochastic rigidity to align frames in a video taken by a single panning camera. The work of Vidal et al. [21] extended the dynamic texture model using time varying linear dynamical system, and proposed a method to calculate the optical flow of non-rigid scenes viewed by a single moving camera. Rav-Acha et al. [16] 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. [1] extended the concept of video textures [17] 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. The results in this paper are very impressive. However the authors acknowledge the fact, that by using techniques for registering non-rigid scenes they could reduce the user intervention for the panoramic video generation. Similar to the panoramic video texture, Rav-Acha et al. [15] proposed the concept of dynamosaicing. They address the problem by finding the registration parameters as the minimal cut in a 4-D graph using max-flow methods.

Caspi et al. have a series of papers [3, 4] that address the problem of spatial-temporal alignment. In [3], two algorithms were proposed - a feature based and a gradient based. The features are extracted using either a KLT tracker [18] 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 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 [4], 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 [20], 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 done even if the two sequences are captured at different times and places. A similarity measure is proposed for each space-time sub-volume. The space time alignment is then obtained by maximizing a similarity measure across all the sub-volumes.

2 Dynamic Textures

For the sake of completeness, in this section we briefly review the dynamic texture model proposed in [5]. This model will be the basis of our registration framework, which we will discuss in the next section.

2.1 Modeling

Given a video sequence, $I(t), t = 1, \ldots, F$, we model it as the output of a linear dynamical system (LDS) of order n. The equations of the model are given by

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

$$I(t) = C_0 + Cz(t) + w(t), \tag{2}$$

where $z(t) \in \mathbb{R}^n$ is a hidden state, $v(t) \sim \mathcal{N}(0, Q)$ is the driving noise, and $w(t) \sim \mathcal{N}(0, R)$ is the measurement noise. The parameters of the model are $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^n$, $C_0 \in \mathbb{R}^p$, $C \in \mathbb{R}^{p \times n}$, where p is the number of pixels in the image. This model essentially decouples the non-rigid object into 2 components - the appearance and the dynamics. The appearance is modeled by the temporal mean of the video C_0 and the matrix C, while the dynamics are modeled by the matrix A.

There are several advantages of using the dynamic texture model. First, since the model is generative, given a finite duration video clip we can learn the parameters of the model and then generate a sequence with the desired number of frames. This property is useful to offset the temporal misalignment in the sequences and generate sequences with the desired number of frames. Second, the non-rigid dynamics of the scene can be modified by changing the eigenvalues of the A matrix, as suggested in [6]. This helps us generate novel sequences with different speed, reversed time evolution, etc.

2.2 Identification

The identification of LDSs can be done using standard system identification methods, such as Subspace Identification N4SID [14]. However, in the case of a dynamic texture the dimension of the output space is equal to the number of pixels, thus using N4SID is very computationally expensive. In this paper we will use the PCA-based approach proposed in [5], which is fast and computationally inexpensive compared to N4SID. The PCA-based approach exploits the fact that when $v = w = 0$, the mean-subtracted video sequence can be stacked into a matrix $W \in \mathbb{R}^{p \times F}$, which can be written as

$$W = [I(1) - \hat{C}_0, \cdots, I(F) - \hat{C}_0] = C[z(1), \cdots, z(F)] = CZ, \tag{3}$$

where $\hat{C}_0 = \frac{1}{F}\sum_{t=1}^{F} I(t)$. Since rank$(W) = n \ll \min\{F, p\}$, one can learn C and Z from the singular value decomposition (SVD) of $W = U\Sigma V^\top$ as $C = U(:, 1\!:\!n)$ and $Z = \Sigma(1\!:\!n, 1\!:\!n)V(:, 1\!:\!n)^\top$. Given Z, solving for A (still assuming that $v = 0$) is a linear problem. Given A and Z, solving for B (assuming now that $v \neq 0$) is also a linear problem. We refer the reader to [6] for the details of the identification algorithm.

2.3 Change of Basis Issue

It is a very well known fact that, every linear dynamical system is unique up to a similarity transformation, i.e. given a linear system (A, B, C, C_0, z_0) and a matrix $S \in \mathcal{S} = GL(n)$, the system given by $(S^{-1}AS, S^{-1}B, CS, C_0, S^{-1}z_0)$ produces the same output, as that of the original system. Therefore, the model parameters are not unique: they are only defined up to a change of basis.

The choice of the basis is irrelevant for synthesis purposes, because the output $I(t)$ is the same irrespective of the basis. But if one wants to compare the model parameters of two systems, then the choice of the basis plays a very important role. Vidal et al. in [21] briefly suggested diagonalizing the A matrix and using this basis for all the systems. However, no evidence was presented in the paper that the Diagonal Canonical Form (DCF) is the best reference basis to use. In fact, a problem with the DCF is that the transformed matrix C is complex whenever A has complex eigenvalues.

To overcome this issue, in this paper we propose to use the Reachability Canonical Form (RCF). The reachability canonical form of the dynamics matrix A is given by

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & -a_3 & \dots & -a_n \end{bmatrix}. \tag{4}$$

where $P(\lambda) = \lambda^n + a_1\lambda^{n-1} + \dots a_n$ is the characteristic polynomial of A.

Given a system (A, B, C, C_0, z_0) of order n, to convert it into RCF we first form the matrix $O = [B, AB, \cdots, A^{n-1}B]^{-1} \in \mathbb{R}^{n \times n}$ and define the row vector q as the nth row of O. The transformation S that converts the system (A, B, C, C_0, z_0) into the RCF $(S^{-1}AS, S^{-1}B, CS, C_0, S^{-1}z_0)$ can now be obtained as

$$S = \begin{bmatrix} q \\ qA \\ \vdots \\ qA^{n-1} \end{bmatrix}^{-1} \in \mathbb{R}^{n \times n} \tag{5}$$

Once the system has been transformed into RCF the bases images are unique up to a sign. However, for purposes of our current application, the sign of the basis becomes redundant due to the way we define the dynamic appearance image in section 2.3. Hence we ignore the sign issues. Once all the systems are in the same reference frame, comparison between parameters is more meaningful.

3 Registration of Non-Rigid Dynamical Scenes

In this section we propose our registration algorithm for non-rigid dynamical scenes. In §3.1 we show how each one of the video registration problems considered in this paper can be reduced to an image registration problem. In §3.2 we show how to register the appearance matrices.

3.1 Reducing the Video Registration Problem to an Image Registration Problem

Recall that the appearance of the video is represented by the matrix $C \in \mathbb{R}^{p \times n}$ and the temporal mean of the video $C_0 \in \mathbb{R}^p$, where p is the number of pixels. 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 C_0 reshaped as an image. With this notation, the dynamic texture model can be rewritten as

$$I(\mathbf{x}, t) = \sum_{i=0}^{n} z_i(t) C_i(\mathbf{x}) + w(t), \tag{6}$$

where $z_0(t) = 1$. Therefore, under the dynamic texture model, a non-rigid dynamical scene is interpreted as a linear combination of n basis images and a mean image.

(a) Multiple synchronized cameras capturing non-rigid scenes. Let $T \in SO(2)$ be a rigid-body transformation, and consider the following two systems

$$z(t+1) = Az(t) + Bv(t) \qquad I(\mathbf{x}, t) = \sum_{i=0}^{n} z_i(t) C_i(\mathbf{x}) + w(t) \tag{7}$$

$$z(t+1) = Az(t) + Bv(t) \qquad \tilde{I}(\mathbf{x}, t) = \sum_{i=0}^{n} z_i(t) C_i(T(\mathbf{x})) + w(t). \tag{8}$$

We see that $\tilde{I}(\mathbf{x}, t) = I(T(\mathbf{x}), t)$. This shows that whenever we apply a constant rigid-body transformation to all the frames in a video sequence, the transformed video has the same A and B matrices as the original video. The main difference is that the appearance images $C_i(\mathbf{x})$ are transformed by *the same* rigid-body transformation.

Therefore, given two synchronized video sequences $I(t)$ and $\tilde{I}(t)$, we can register them as follows. We can first learn the system parameters for the two video sequences and transform them into RCF. Let (A, B, C, C_0, z_0) and $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{C}_0, z_0)$ be the parameters of the two systems. In the ideal case, we will have $A = \tilde{A}$ and $B = \tilde{B}$ (up to sign). Also, since the cameras are synchronized, both systems start from the same phase of the dynamics and so we will have $z_0 = \tilde{z}_0$ (up to sign). The registration of the videos is then obtained by registering the appearance images $C_i(\mathbf{x})$ and $\tilde{C}_i(\mathbf{x})$ using any image registration technique, as we will discuss in §3.2.

(b) Multiple unsynchronized cameras capturing non-rigid scene. In the synchronized case, one could argue that the registration of the two video sequences could have been done by registering a pair of frames from the two video sequences, rather than using the dynamic texture model. However, when the cameras are not synchronized, registering corresponding frames of the video would give poor registration results. To make the frame-to-frame registration work, the temporal correspondences need to be first determined and then the images need to be spatially aligned.

Thanks to the dynamic texture model, the lack of temporal synchronization between the two videos does not cause any change to the algorithm for the synchronized case.

This is because the effect of a temporal shift on the videos is a temporal shift on the hidden state $z(t)$, without affecting the appearance images. More specifically, if the two videos are related by a spatial transformation T and a temporal shift τ, i.e. $\tilde{I}(\mathbf{x}, t) = I(T(\mathbf{x}), t + \tau)$, then the appearance images are related by the spatial transformation only, i.e. $\tilde{C}_i(\mathbf{x}) = C_i(T(\mathbf{x}))$, and the hidden states are related by the temporal shift only, i.e. $\tilde{z}(t) = z(t + \tau)$. Therefore, we may register the two videos by registering the appearance images, irrespective of any temporal lag.

Once the sequences are spatially aligned there are several choices to temporally align them. One of the methods is to do a 1-dimensional search in the temporal direction and pick the temporal offset that gives the least error between the two sequences, after applying the recovered spatial transformation to it. The temporal offset τ is given by

$$\tau = \underset{\tau}{\operatorname{argmin}} \sum_{t=1}^{F-\tau} \|\tilde{I}(\mathbf{x}, t) - I(T(x), t + \tau)\|^2. \tag{9}$$

A simpler method is to simply align the hidden states of the two systems, i.e.

$$\tau = \underset{\tau}{\operatorname{argmin}} \sum_{t=1}^{F-\tau} \|\tilde{z}(t) - z(t + \tau)\|^2. \tag{10}$$

However, this simpler method does not perform as well, because the hidden states need to be estimated. Therefore, in this paper we use the first method to temporally align the sequences.

(c) Single moving camera capturing a non-rigid scene. When the camera is moving, the dynamic texture model is no longer valid, because it uses a constant appearance matrix, while in reality the appearance changes due to the camera motion. To deal with this issue, Vidal et al. [21] proposed to identify a time-varying dynamic textures model

$$z(t + 1) = Az(t) + Bv(t) \tag{11}$$

$$I(t) = C(t)z(t) + w(t). \tag{12}$$

Echoing the same notion that $C(t)$ captures the appearance, they proposed a dynamic texture constancy constraint on $C(t)$ from which one can compute the optical flow of a dynamic texture seen by a moving camera. Since identification of time-varying systems is not straightforward, the method in [21] estimates $C(t)$ by applying the PCA-based method described in §2.2 to a moving window of frames.

We use a similar approach in this paper. Given a sequence $I(t), t = 1, \ldots, F$, we slide a temporal window of size τ to obtain $F - \tau$ subsequences. We then register each pair of consecutive subsequences using their appearance images, as described before. We then create a panoramic image using the first frame of each registered subsequence.

3.2 Registering the Appearance Images

In the previous section we showed how the registration of different kinds of video sequences can be reduced to the problem of spatially registering their appearance images. In this section we show how to perform the actual registration of the appearance images.

The columns of the C matrix represent the principal components of the appearance and we want to account for the contribution of each component. To that end, we first scale the basis images according to their singular values. By doing this, we retain the contribution of the each basis image to the appearance of the non-rigid scene. We then sum each of the scaled basis images to obtain the *dynamic appearance image* $C_a(\mathbf{x})$. While doing so we do not want regions canceling out each other, otherwise the appearance of the object will be lost. Therefore, we take the sum of the absolute values of the scaled basis images, i.e.

$$C_a(\mathbf{x}) = \sum_{i=1}^{n} |\sigma_i C_i(\mathbf{x})| \tag{13}$$

Given two appearance matrices C and $\tilde{C}$, we first compute their dynamic appearance images $C_a(\mathbf{x})$ and $\tilde{C}_a(\mathbf{x})$ respectively. Then, we can register the two video sequences by applying any standard image registration to the following images

1. Registration based on the temporal means: $C_0(\mathbf{x})$ and $\tilde{C}_0(\mathbf{x})$.
2. Registration based on the dynamic appearance images: $C_a(\mathbf{x})$ and $\tilde{C}_a(\mathbf{x})$.
3. Registration based on both: $C_a(\mathbf{x}) + C_0(\mathbf{x})$ and $\tilde{C}_a(\mathbf{x}) + \tilde{C}_0(\mathbf{x})$.

As we will see in §4, which one of the three methods performs best in practice depends on the sequence.

Algorithm 1 summarizes our spatial temporal registration approach.

Algorithm 1 Method for registering static video sequences

```
1. Given $I(t)$ and $\tilde{I}(t)$ identify the model parameters $(A, B, C, C_0, z_0)$ and $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{C}_0, \tilde{z}_0)$.
2. Compute the transformations $S$ and $\tilde{S}$ that convert the models into RCF and set $C = CS$ and $\tilde{C} = \tilde{C}\tilde{S}$.
3. Compute $C_a(\mathbf{x})$ and $\tilde{C}_a(\mathbf{x})$.
4. Spatially register the dynamic appearance images formed from both the two sequences (either $C_0$, or $C_a$, or $C_0 + C_a$) using an image registration algorithm.
5. Temporally register the spatially registered videos using 1-D search in the temporal direction.
```

4 Experimental results

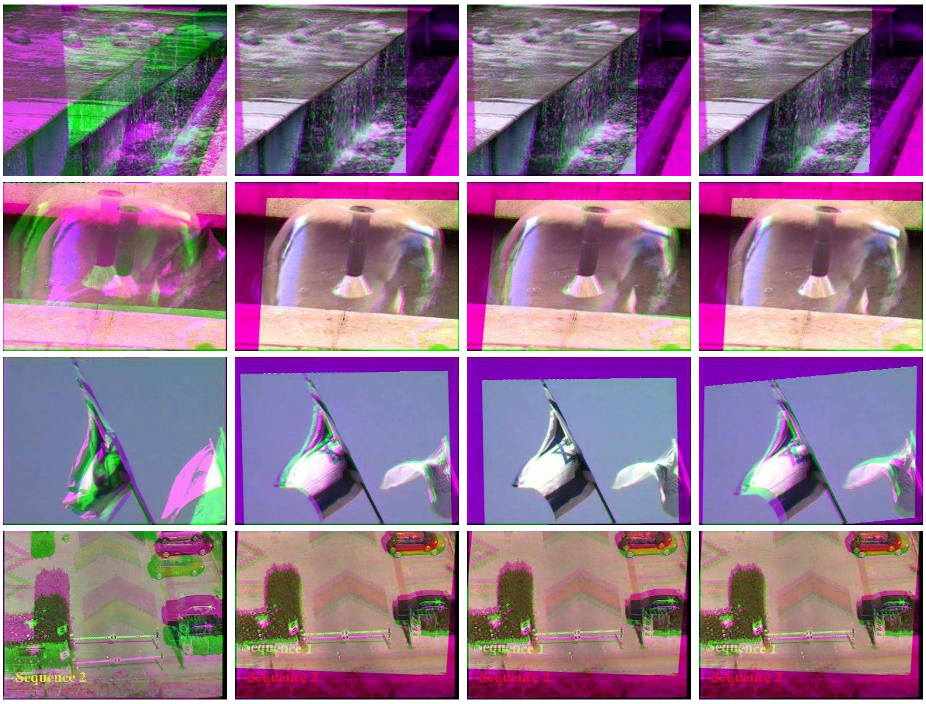
In this section, we present video registration results for the different cases outlined in the previous section. In particular, we present results for the case of multiple unsynchronized videos seen by a static camera and the case of a single video seen by a moving camera. In the unsynchronized case, we also compare our results to those of the algorithm in [3]. The case of multiple synchronized videos is the easiest of the three cases and for lack of space we do not present results for that case.

As stated earlier we use the PCA-based suboptimal method for the identification of the dynamic texture parameters. For the image registration, we use the MATLAB code [7]. This algorithm allows for local deformations too, but we only utilize the global registration framework provided by this algorithm. Since the procedure outlined in this paper reduces the video alignment case to an image registration problem, one could use any image registration algorithm and apply our framework. The accuracy of the results obtained will be affected by the accuracy of the registration algorithm.

Figure 1 shows results on registration of multiple unsynchronized videos. We display the results by superimposing a frame from the first video with the corresponding frame from the second video as follows: the first and third color channels are from the first image and the second color channel is from the second image. The first column shows the superposition of the two images before any registration. The second column shows the registration using the mean image. The third column shows the registration using the dynamic appearance image, and the fourth column shows the registration using both the mean and the dynamic appearance images. The frame of reference is with respect to the first image. In our experiments we manually choose the order of the LDS in the range 50-75 depending on the sequence. However, one could also use model selection to choose the order of the systems.

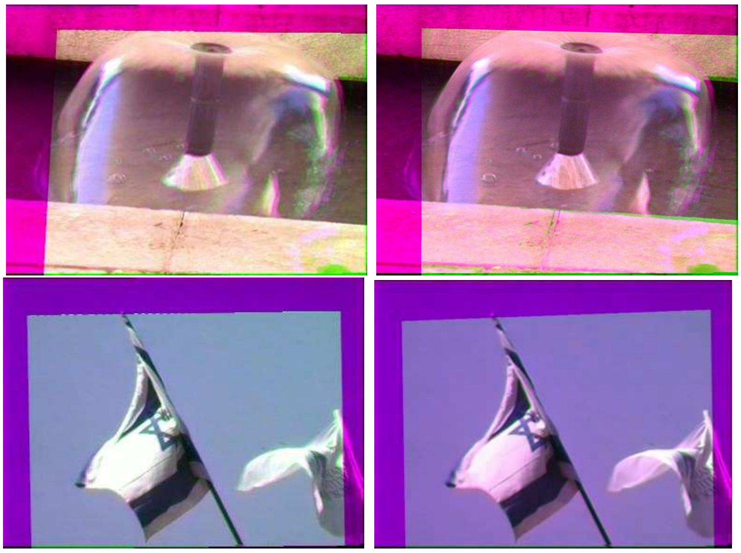
For the wall fountain sequence, good registration results are achieved for all the 3 cases, i.e. using the mean, the dynamic appearance , or both. This is because the sequence has considerable dynamics and at the same time the mean carries enough texture information to use it for registration. For the fountain sequence, using the dynamic appearance images produces good registration. However, notice that using the mean alone provides a better registration than using the dynamic appearance image alone. Thus, the registration is very efficient as one simply needs to calculate the mean to accurately register the sequence. For the flag sequence, using the dynamic appearance image alone produces better results compared to the other two cases. This is because the mean image predominantly consists of the sky, which lacks texture information, giving rise to the well known aperture-problem. Although our algorithm is tailored to non-rigid scenes, we see that the application of the algorithm to rigid scenes also works reasonably well. For the parking lot sequence, we see that the registration results are not as accurate as in the other three sequences.

In Figure 2 we compare the results of our methods to the results presented in [3]. We see that the results from our method are as good as the results from the earlier work. However, our method is simpler, computationally less intensive and does not rely on any tracking of features as compared to [3]. One of the salient features of our method is that, we can recover the spatial registration of the sequences, irrespective of the temporal alignment of the sequences.



(a) Initial alignment (b) Using mean (c) Using appearance (d) Using both

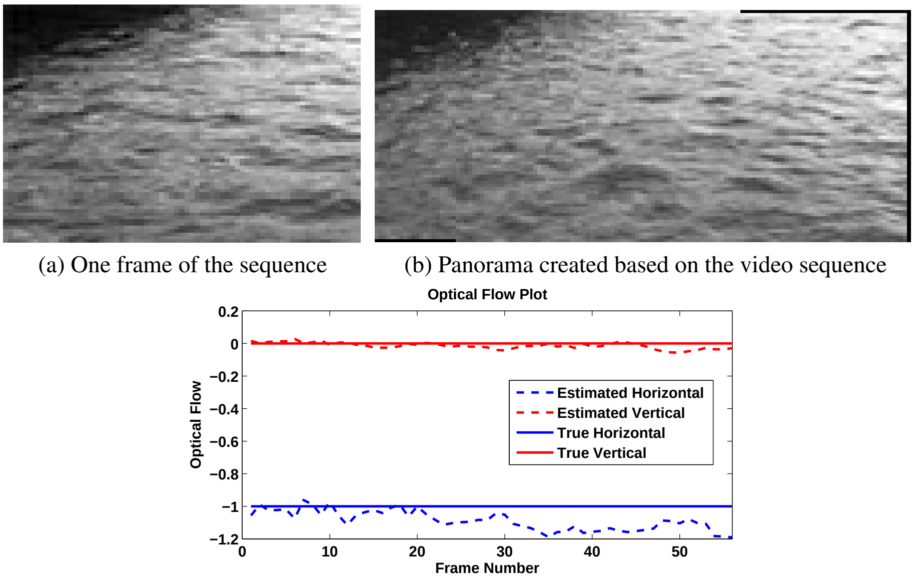
Fig. 1. Spatial-temporal registration results for the different sequences. **First row:** Wall fountain sequence ($n = 55$). **Second Row:** Fountain Sequence ($n = 50$). **Third Row:** Flag Sequence ($n = 55$). **Fourth Row:** Fountain Sequence ($n = 75$).



(a) Our Method (b) Results from [3]

Fig. 2. Comparison of results from our method to the results from [3].

Figure 3 shows registration results for the moving camera case. Here we generate a synthetic sequence that follows the time-varying dynamic texture model. The main reason for doing this is that tracking points in non-rigid dynamical sequences is very hard. Figure 3 shows the plot of the estimated optical flow vs the true optical flow. This graph shows us that the algorithm can be extended to the moving camera case as well. This scene is totally non-rigid and it will be very difficult to extract good point trajectories from these sequences to compare them. Figure 3 shows one frame of the sequence and the panoramic image made after registering the entire sequence. Running traditional methods on such sequences would give poor results.



(a) One frame of the sequence

(b) Panorama created based on the video sequence

(c) Optical flow plot

Fig. 3. Registration of moving camera on non-rigid scene

5 Conclusions and future work

We have established that registering video sequences based on the dynamic texture framework helps us to obtain a temporally invariant spatial registration algorithm. The choice of whether one uses the mean, the dynamic appearance image or both depends on the sequence and one can decide this based on the application. Our approach is very straightforward and efficient. It does not need to track trajectories or feature points. It also gives an interesting flavor by reducing video registration problems to an image registration problem. This gives us access to a large number of tools from the image registration community.

Future work includes exploring the properties of the dynamics of the system to create interesting and novel sequences. Automatic model selection for the system parameters is another direction of future work.

Acknowledgments

The authors thank Rizwan Chaudhry for his help with the data collection and experiments. This work has been funded with startup funds from the Whiting School of Engineering of The Johns Hopkins University, by grants ONR N00014-05-10836, NSF CAREER IIS-0447739, and NSF EHS-0509101.

References

1. A. Agarwala, K. C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski. Panoramic video textures. *SIGGRAPH*, 24(3):821–827, 2005.
2. Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):120–135, 2001.
3. Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002.
4. Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006.
5. G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.
6. G. Doretto and S. Soatto. Editable dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 137–142, 2003.
7. H. Farid and S. Periyaswamy. Image registration. http://www.cs.dartmouth.edu/farid/research/registration.html.
8. M. A. Fischler and R. C. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26:381–395, 1981.
9. A. Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. In *IEEE International Conference on Computer Vision*, pages 662–669, 2001.
10. C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, 1988.
11. V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics, SIGGRAPH 2003*, pages 277–286, 2003.
12. I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision*, 2003.
13. D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.
14. B. D. Moor, P. V. Overschee, and J. Suykens. Subspace algorithms for system identification and stochastic realization. Technical Report ESAT-SISTA Report 1990-28, Katholieke Universiteit Leuven, 1990.
15. A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaics: Video mosaics with non-chronological time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 58–65, Washington, DC, USA, 2005. IEEE Computer Society.

16. A. Rav-Acha, Y. Pritch, and S. Peleg. Online registration of dynamic scenes using video extrapolation. In *Workshop on Dynamic Vision (ICCV)*, 2005.
17. A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *SIGGRAPH*, pages 489–498, 2000.
18. J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.
19. M. Szummer and R. W. Picard. Temporal texture modeling. In *IEEE International Conference on Image Processing*, volume 3, pages 823–826, 1996.
20. Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *European Conference on Computer Vision*, pages 538–550, 2006.
21. R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 516–521, 2005.
22. P. A. Viola. Alignment by maximization of mutual information. Technical Report AITR-1548, 1995.
23. L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH*, 2000.