View-invariant Alignment and Matching of Video Sequences

PaperID 440

Abstract

In this paper, we propose a novel method to establish temporal correspondence between the frames of two videos. The 3D epipolar geometry is used to eliminate the distortion generated by the projection from 3D to 2D. Although the fundamental matrix contains the information of the extrinsic property of projective geometry between views, it is sensitive to noise and fails for non-rigid human body movement. Therefore, we propose the use of a rank constraint of the corresponding points in two views to measure the similarity between trajectories. This rank constraint shows more robustness and is easier to compute than the fundamental matrix. Furthermore, a dynamic programming approach using the similarity measurement is proposed to find the non-linear time-warping function for videos containing human activities. In this way, videos of different individuals taken at different times and from distinct viewpoints can be synchronized. Moreover, a temporal pyramid of trajectories is applied to improve the accuracy of the view-invariant dynamic time warping approach. We show various applications of this approach, such as video synthesis, human action recognition and computer aided training. Compared to the state-of-the-art techniques, our method shows a great improvement.

1. Introduction

Many applications, such as video mosaicing, video retrieval, image based modelling and rendering, video synthesis, multi-sensor surveillance, and human action recognition, require a computation of a spatio-temporal alignment of video sequences. Methods that tackle this problem discover a correspondence between the video sequences. Some of these methods assume the input video sequences are already synchronized, while the other methods use an optional built-in expensive hardware that provides synchronization. This paper presents a novel approach of alignment and matching of video sequences. We only assume that given two video sequences are correlated due to the motion of objects. Based on this correlation we discover the correspondences (temporal alignment) between the frames of one sequence to other.

When a feature point moves in a 3D space with respect to time, it generates a 3D trajectory: $\{(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), \ldots, (X_t, Y_t, Z_t)\}$, where t is the time stamp. This 3D trajectory is projected as a 2D trajectory in the image plane: $\{(u_1, v_1), (u_2, v_2), \ldots, (u_t, v_t)\}$. The relationship between a point (X_i, Y_i, Z_i) in 3D, trajectory and its 2D projection (u_i, v_i) is defined as follows:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = P \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, i = 1, 2, ..., t, \quad (1)$$

where P is the projection matrix (camera model).

Assume that the same motion is performed with a different speed (temporal extent), then we obtain another 3D trajectory: $\{(X_{C(1)}, Y_{C(1)}, Z_{C(1)}), (X_{C(2)}, Y_{C(2)}, Z_{C(2)}), \cdots, (X_{C(t)}, Y_{C(t)}, Z_{C(t)})\}$,where $C(i)$ is a time warping function such that

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} X_{C(i)} \\ Y_{C(i)} \\ Z_{C(i)} \\ 1 \end{bmatrix}, i = 1, 2, ..., t$$

Now assume that the viewpoint of the camera has also been changed. Then the projection of this 3D trajectory to a 2D trajectory, $\{(u'_1, v'_1), (u'_2, v'_2), \ldots, (u'_t, v'_t)\}$, is defined in the similar way:

$$\begin{bmatrix} u'_{C(i)} \\ v'_{C(i)} \\ 1 \end{bmatrix} = P' \begin{bmatrix} X_{C(i)} \\ Y_{C(i)} \\ Z_{C(i)} \\ 1 \end{bmatrix}, i = 1, 2, ..., t.$$

Therefore, the problem of aligning video sequences is to discover the time-warping function, $C(i)$, for $i = 1, 2, \ldots, t$, using the information in two 2D trajectories, $\{(u_1, v_1), (u_2, v_2), \ldots, (u_t, v_t)\}$ and $\{(u'_{C(1)}, v'_{C(1)}), (u'_{C(2)}, v'_{C(2)}), \ldots, (u'_{C(t)}, v'_{C(t)})\}$.

There are two crucial aspects of exploring correspondences of video sequences. First, the 2D trajectory is highly dependent on the viewpoint, that is the same 3D trajectory may look different in videos shot from the different viewpoints. Second, the same motion may have different speeds (temporal extents). The second problem becomes more complicated when the motion changes dynamically, such that the indices of corresponding frames are non-linearly related. This is very common in videos depicting human activities, since even the same person may perform the same activity with different speeds. In this paper, we propose a novel approach for alignment and matching of videos, which is based on the epipolar geometry and can discover the complex time-warping function, $C(t)$.

There are two main types of approaches for aligning sequences: sequence-to-sequence and trajectory-to-trajectory. The sequence-to-sequence approach, which is also called direct approach, takes the video frames as an input and applies the computation over all pixels in the video frames. The trajectory-to-trajectory approach tracks the movement of the feature points in the field of view, and computation is based on the information from the trajectories. The advantages of direct approach include: it determines more accurately the spatial transformation between sequences than trajectory-to-trajectory approach, and it does not require explicit feature detection and tracking. On the contrary, since the trajectories contain explicit geometric information, the trajectory-to-trajectory approach better determines the large spatio-temporal misalignments, can align video sequences acquired by different sensors and is less affected by the background changes. The detailed comparison between these approaches is available in [13, 1]. Since the video sequences

in most applications contain a significant spatio-temporal variance, we choose trajectory-to-trajectory approach. As one of the achievements, based on the trajectory information we can align the video sequences where different people perform the same action.

Previously, researchers have tried using calibrated/uncalibrated stereo-rigs [6, 8] to recover the projection relationships among the videos. In these approaches, the fundamental matrix is used to find the spatial relationship between the trajectories [3, 14]. However, due to the instability of reconstruction process, those approaches can only be applied to some limited video sequences, such as videos simultaneously shot. Therefore, there is no previous methods to synchronize two videos of different people performing the same 3D activity at different time using the fundamental matrix.

In this paper, we propose a method, which is based on the epipolar constraint, but does not need explicit reconstructing of the 3D relationships. This method can align videos containing 3D actions with large spatio-temporal variance. Since it is a well-studied problem to reconstruct the spatial alignment of video sequences given the correspondent frames, we do not discuss the spatial registration. The results of experiments show that our method is much more stable, and it can be used in many applications.

1.1. Previous Work

Stein [11] achieved the alignment of tracking data obtained from multiple camera assuming homography relationship between the cameras. Stein did not use the trajectory information, but discovered the temporal alignment using exhaustive search among different intervals between video sequences. Due to this, his method computationally quite expensive, and it can only align the videos with a constant time shift.

Giese and Poggio [7] proposed a method to find the spatio-temporal alignment of two video sequences using the dynamic shift of the time stamp of the spatial information. They assumed that a 2D action trajectory can be represented as a linear-combination of prototypical views, and the effect of viewpoint changes can be expressed by varying the coefficients of the linear-combination. Since they did not use the 3D information, this method can only align some simple motion patterns.

Caspi and Irani [1] proposed a direct approach to align two surveillance videos by finding the spatio-temporal transformation that minimizes the sum of squares differences (SSD) between the sequences. They extended the direct approach to the alignment of non-overlapping sequences captured by a stereo rig [2]. In these video sequences, the same motion induces "similar" changes in time. This correlated temporal behavior was used to recover the spatial and temporal transformations between sequences. They also proposed a trajectory-to-trajectory approach for alignment of sequences captured by cameras with significant different viewpoints [3]. In this method the alignment of trajectories is based on computation of the fundamental matrix. Their approaches can only be used for applications, in which the time shift between the video sequences is constant or is a linear function. Therefore, their method will fail for videos with a dynamic time shift.

Wolf and Zomet [14] proposed a method for self calibrating a moving rig. During the movement, the viewing angles between cameras and the time shift are fixed, but the internal camera parameters are allowed to change.

Extensive research has been done for action recognition, various approaches were applied to discover the viewpoint difference between videos, to measure the difference between actions using view-invariant characteristics, or to find the period of the cyclic motion [12, 9, 10].

From these reviews, we can conclude that the existent methods are not appropriate for alignment of video sequences containing the complex 3D motion with significant spatio-temporal expansion.

2. View-invariant Alignment of Video

We propose a dynamic computation of time-warping function, and a novel measurement of similarity that is based on epipolar geometry.

2.1. View-invariant Measure

First, let us consider the measuring similarity between 2D trajectories, which are represented as $\{(u_1, v_1), (u_2, v_2), \ldots, (u_t, v_t)\}$ and $\{(u'_{C(1)}, v'_{C(1)}), (u'_{C(2)}, v'_{C(2)}), \ldots, (u'_{C(t)}, v'_{C(t)})\}$.

In the Eq. 1, the general camera projection can be modeled using the following perspective matrix

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{23} & p_{33} & p_{34} \end{bmatrix}.$$

Readers could reference to any computer vision textbook to find the properties of this projection matrix. In this paper, we focus on the epipolar geometry, which represents the extrinsic projective geometry between views.

For the perspective model, the fundamental matrix (a 3 by 3 matrix), $\mathbf{F}$, is defined by the equation

$$s(i) = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}^T \mathbf{F} \begin{bmatrix} u'_{C(i)} \\ v'_{C(i)} \\ 1 \end{bmatrix} = 0, \tag{2}$$

for a pair of matching points $(u_i, v_i) \leftrightarrow \left(u'_{C(i)}, v'_{C(i)}\right)$ in two trajectories. Therefore, given a fundamental matrix, we can use Eq. (2) to measure the similarity between trajectories, such that the summation of $s(i)$ for all points is minimized.

It is a well known fact that the computation of fundamental matrix is not robust. The non-rigid motion can further worsen the stability. Example of non-rigid motion includes video sequences containing human activities captured at different time. In other words, two sequences of two different people performing the same action captured by the same camera from two different viewpoints. If a person performs the same movement differently, and the motion trajectories are non-rigid, previous approaches [3, 14] will fail to synchronize these two video sequences. Therefore, we propose a novel approach, which avoids the computation of the fundamental matrix.

Given sufficiently many point matches, Eq. (2) can be used to compute the unknown matrix F from the following equation:

$$\mathbf{Mf} = \begin{bmatrix} u'_{c(1)}u_1 & \cdots & u'_{c(t)}u_t \\ u'_{c(1)}v_1 & \cdots & u'_{c(t)}v_t \\ u'_{c(1)} & \cdots & u'_{c(t)} \\ v'_{c(1)}u_1 & \cdots & v'_{c(t)}u_t \\ v'_{c(1)}v_1 & \cdots & v'_{c(t)}v_t \\ v'_{c(1)} & \cdots & v'_{c(t)} \\ u_1 & \cdots & u_t \\ v_1 & \cdots & v_t \\ 1 & \cdots & 1 \end{bmatrix}^T \mathbf{f} = 0 \quad (3)$$

where f is the rearrangement of the elements of the fundamental matrix: $f = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{21} & f_{22} & f_{23} & f_{31} & f_{32} & f_{33} \end{bmatrix}^T$. Let us denote the $\mathbf{M}$ by the observation matrix, which is constructed using the coordinates of points of two 2D trajectories. Since (3) is a homogenous equation, for a solution of $\mathbf{f}$ to exist, matrix $\mathbf{M}$ must have rank at most eight. However, due to the noise or the matching error, the rank of matrix $\mathbf{M}$ may not be exactly eight. In this case the 9^{th} singular value of $\mathbf{M}$ estimates the necessary perturbation of coordinates of each point in matrix $\mathbf{M}$ to produce two projections of the same 3D trajectory. Therefore, we can use the 9^{th} singular value of matrix $\mathbf{M}$ to measure the matching of two trajectories. The smallest singular value of $\mathbf{M}$ corresponds to the best match of trajectories, and we denote it as dis.

We generated two trajectories, selected nine points from each trajectory and put them into the observation matrix $\mathbf{M}$. The 9^{th} eigenvalue increases dramatically when there is a large change in the x and y coordinates of one point, and it is close to zero only within a very small range. Therefore, if the points are spread far enough from each other (that is the points are not clustered in one specific location), by picking the nine corresponding points from each trajectory, we can decide whether two trajectories match or not. Moreover, since the trajectory contains the temporal information, we can also use this temporal information to align trajectories. We discuss the use of temporal information for alignment in the section 2.2.

In some applications it is reasonable to assume that the time-warping function is linear, $C(i) = ai + b$. Then a and b parameters of the time-warping function, can be found by using the exhaustive search and by minimizing the $dist$ measures. And to model more complicated time-warping functions, a higher order polynomial must be used. However, these types of time-warping function have very limited applications, such as synchronizing two video sequences that are captured simultaneously, or synchronizing stereo cameras. Generally, this approach fails to align video sequences shot at different times and contain human activities, since the time-warping function for human activities can not be modelled by a polynomial.

2.2. View-invariant Dynamic Time Warping

Dynamic Time Warping (DTW) is a widely used method for warping two temporal signals. It uses an optimum time expansion/compression function to perform non-linear time alignment (See Figure. 2.2). The applications include speech recognition, gesture recognition, signature recognition [5]. For two signals I and J, a distance measure E is computed to measure the misalignment between the temporal signals, where $E(i, j)$ represents the error of aligning signals (distance measure) up to the time instants t_i and t_j respectively. It is The error of alignment

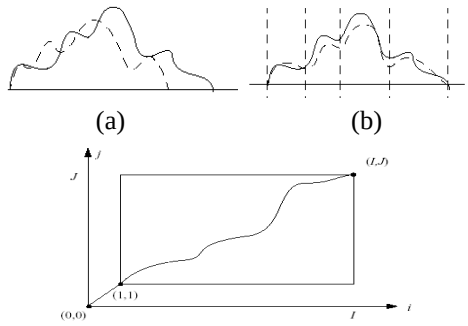


Figure 1: a) Two temporal signals, b) after time warping, c) the distance metric C and the warping path.

is computed incrementally using the formula:

$$\begin{aligned} &E(i,j) = d_{i,j} + e, and \\ &e = \min\{E(i-1,j), E(i-1,j-1), E(i,j-1)\} \end{aligned} \quad (4)$$

Here d_{ij} captures the cost of making time instants t_i and t_j correspond to each other. The best alignment is then found by keeping track of the elements that contribute the minimal alignment error at each step and backward following a path from element $E(i, j)$ to $E(1, 1)$.

The above method can only align video sequences shot from the same viewpoint. To achieve the view-invariance, we introduce the 3D shape information into the analysis through the d_{ij} distance measure. Based on the view-invariant similarity metric in Section 2.1, we propose a view-invariant DTW algorithm as follows:

(1) We specify eight corresponding points between the first frames of two videos, and denote the image coordinates as $(x'_1, y'_1), ..., (x'_8, y'_8)$ and $(x_1, y_1), ..., (x_8, y_8)$.

(2) Track the feature points in two videos to acquire trajectories I=$\{(u'_1, v'_1),\ldots,(u'_n, v'_n)\}$ and J=$\{(u_1, v_1),\ldots,(u_m, v_m)\}$. In our experiments we used the mean-shift tracker [4].

(3) For each pair of the corresponding points in the trajectories, construct the 9×9 observation matrix:

$$M_O = \begin{bmatrix} x'_1x_1 & \cdots & x'_8x_8 & u'_iu_j \\ x'_1y_1 & \cdots & x'_8y_8 & u'_iv_j \\ x'_1 & \cdots & x'_8 & u'_i \\ y'_1x_1 & \cdots & y'_8x_8 & v'_iu_j \\ y'_1y_1 & \cdots & y'_8y_8 & v'_iv_j \\ y'_1 & \cdots & y'_1 & v'_i \\ x_1 & \cdots & x_8 & u_j \\ y_1 & \cdots & y_8 & v_j \\ 1 & \cdots & 1 & 1 \end{bmatrix}^T . \quad (5)$$

(4) Execute the classic DTW algorithm but using the distance measure between the points at t_i and the t_j respectively of two trajectories: $d_{(i,j)} = dis(i, j)$, where $dis(i, j) = \sigma_9$ is the 9^{th} singular value of the matrix $\mathbf{M_O}$ in step 3.

(5) Generate the time-warping function $C(i) = i, i = 1, \ldots, n$ by back tracing the path that minimize the value of $E(i, j)$ from the upper-left corner of the matrix $\mathbf{E}$. If the cell $E(i, j)$ is on the warping path, it means i^{th} point of trajectory I corresponds to the j^{th} point of trajectory J.

Note that the DTW can establish the correspondence "on the fly", which means that it determines the best warping path to

element $E(i,j)$. To achieve more robust measurement for the $E(i,j)$, we put the previously found corresponding points up to i and j in the observation matrix $\mathbf{M}$, and update the original observation matrix $\mathbf{M_O}$ Eq. 2.2. The matrix $\mathbf{M_R}$ is given as follows:

$$M_R = \left[\frac{M_P}{M_O}\right]; \quad M_P = \begin{bmatrix} u'_1u_1 & \cdots & u'_{i-1}u_{j-1} \\ u'_1v_1 & \cdots & u'_{i-1}v_{j-1} \\ u'_1 & \cdots & u'_{i-1} \\ v'_1u_1 & \cdots & v'_{i-1}u_{j-1} \\ v'_1v_1 & \cdots & v'_{i-1}v_{j-1} \\ v'_1 & \cdots & v'_{i-1} \\ u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \\ 1 & \cdots & 1 \end{bmatrix}^T \quad (6)$$

This algorithm is not affected by the change in the viewpoint, since the matching measure does not depend on the viewpoint, and it dynamically computes the non-linear time-warping function between the two 2D trajectories.

2.3. Temporal Coarse-to-fine refinement

As we mentioned in section 2.1, the matching measure does not require the explicit computation of the fundamental matrix, therefore the $rank(\mathbf{M}) = 8$ is only a necessary condition to determine whether the two points match or not. It can be noticed that the last singular value of the observation matrix shows an ambiguity if there are many points are very close to the correct one. Therefore, the matching algorithm might give wrong results due to the noise in the trajectory. The DTW is also sensitive to the errors, such that if the warping function is incorrect at $E(i,j)$, then the error will be propagated to the rest of the warping path. To solve these problems we use temporal pyramids of trajectories.

In the temporal pyramid, the higher level has less number of points, and the distance between consecutive points is relatively greater than the one in the lower level. The larger distance between points generates the larger change of the last singular value. Consequently, the significant variation of the last singular value determines matching points without the ambiguity. Furthermore, the higher level of the pyramid provides a constraint for the lower level by propagating point correspondence. So by using the coarse-to-fine approach, we can prevent the error propagation to the rest of the time-warping function.

We propose a novel coarse-to-fine refinement for the view-invariant DTW algorithm:

(1) For the trajectory I use spline to sub-sample the trajectory by factor of 2, such that $length(\mathbf{I}^k) = 0.5 * length(\mathbf{I}^(k+1))$ (length() is the total number of points in the trajectory), where k is the index of level of the pyramid and the highest level is labelled as $k = 0$. The same approach is applied for the trajectory J. And the coordinates of i^{th} point in trajectory I^k is represented as $((u'_i)^k, (v'_i)^k)$, and the j^{th} point in trajectory J^k is represented as $((u_j)^k, (v_j)^k)$.

(2) At the top level $(k = 0)$ compute view-invariant DTW using I^0 and J^0.

(3) For the $k+1$ level, generate the observation matrix, whose first rows are the rows of observation matrix $\mathbf{M}$ from the k level. The matrix $\mathbf{M}$ is arranged as following: $M_R = \left[\frac{M_P}{\frac{M_Q}{M_O}}\right]$

$$M_P = \begin{bmatrix} (u'_1)^k(u_1)^k & \cdots & (u'_{tn})^k(u_{tm})^k \\ (u'_1)^k(v_1)^k & \cdots & (u'_{tn})^k(v_{tm})^k \\ (u'_1)^k & \cdots & (u'_{tn})^k \\ (v'_1)^k(u_1)^k & \cdots & (v'_{tn})^k(u_{tm})^k \\ (v'_1)^k(v_1)^k & \cdots & (v'_{tn})^k(v_{tm})^k \\ (v'_1)^k & \cdots & (v'_{tn})^k \\ (u_1)^k & \cdots & (u_{tm})^k \\ (v_1)^k & \cdots & (v_{tm})^k \\ 1 & \cdots & 1 \end{bmatrix}^T$$

$$M_Q = \begin{bmatrix} (u'_1)^{k+1}(u_1)^{k+1} & \cdots & (u'_{i-1})^{k+1}(u_{j-1})^{k+1} \\ (u'_1)^{k+1}(v_1)^{k+1} & \cdots & (u'_{i-1})^{k+1}(v_{j-1})^{k+1} \\ (u'_1)^{k+1} & \cdots & (u'_{i-1})^{k+1} \\ (v'_1)^{k+1}(u_1)^{k+1} & \cdots & (v'_{i-1})^{k+1}(u_{j-1})^{k+1} \\ (v'_1)^{k+1}(v_1)^{k+1} & \cdots & (v'_{i-1})^{k+1}(v_{j-1})^{k+1} \\ (v'_1)^{k+1} & \cdots & (v'_{i-1})^{k+1} \\ (u_1)^{k+1} & \cdots & (u_{j-1})^{k+1} \\ (v_1)^{k+1} & \cdots & (v_{j-1})^{k+1} \\ 1 & \cdots & 1 \end{bmatrix}^T$$

$$M_O = \begin{bmatrix} x'_1x_1 & \cdots & x'_8x_8 & (u'_i)^{k+1}(u_j)^{k+1} \\ x'_1y_1 & \cdots & x'_8y_8 & (u'_i)^{k+1}(v_j)^{k+1} \\ x'_1 & \cdots & x'_8 & (u'_i)^{k+1} \\ y'_1x_1 & \cdots & y'_8x_8 & (v'_i)^{k+1}(u_j)^{k+1} \\ y'_1y_1 & \cdots & y'_8y_8 & (v'_i)^{k+1}(v_j)^{k+1} \\ y'_1 & \cdots & y'_1 & (v'_i)^{k+1} \\ x_1 & \cdots & x_8 & (u_j)^{k+1} \\ y_1 & \cdots & y_8 & (v_j)^{k+1} \\ 1 & \cdots & 1 & 1 \end{bmatrix}^T .$$

(4) Continue the measurement of matching trajectories I (k+1) and J (k+1).

(5) Repeat steps 3 and 4 till the lowest level.

Thus, the correspondence of points from the upper level is smoothly transitioned to the lower level of the pyramid. The ambiguity is resolved and the error does not affect the rest of time-warping function.

3. Examples and Applications

We have applied our algorithm on various video sequences. First, we used synthetic trajectory data for an accurate evaluation of proposed approach. Next, we apply our method to synchronize the real videos. From Caspi and Irani's experiments [2] we chose sequences acquired by the cameras with non-overlapping FOVs, and the cameras with zoom and no zoom overlapping FOV in order to show the view-invariace of the proposed approach. The alignment of videos, containing human activities captured by moving and stationary cameras, illustrates the robustness of the view-invariant measure used in DTW. The synchronization of the videos of different dancers and matching results can be applied in training dancers. Finally, we applied the algorithm to a long video, containing 60 actions performed by different people, to retrieve automatically similar actions.

3.1. Synthetic Examples

We generated a 3D sinusoidal curve, and projected it onto 2D plane using different projection matrices. Fig. 2 (a) shows the synthetic 3D trajectory, and Fig. 2 (b) shows the projected 2D trajectories.

First, we used the (x, y) coordinates of the trajectories for general DTW algorithm. The DTW using Euclidian distance

	Perspective camera model with rank constraint similarity	Fundamental matrix based similarity
No noise	Fig.3(a): excellent result	Fig.3(a): excellent result
With noise	Same as Fig.3(a): excellent result	Fig.3(b): very bad result

Table 1: The performance evaluation for different model based approaches. Each approach was tested with perfect data and degenerated data.

cannot solve the correspondence at all, since the shape of two trajectories is significantly different due to the projection effects. Second, we compared the view-invariant metric using the rank constraint and applied view-invariant DTW to obtain correspondence. Fig. 2(c)shows the result, such that the dotted lines connect the corresponding points in each trajectory. Table 3.1 shows the error under different conditions.

The noise with a normal distributed with $\sigma = 0.00001$ and $mean = 0$ was added to the 2D trajectories. Fig. 3 shows the histogram of correspondence errors for different methods. In this figure, 0 error represents the correct correspondence result, 1 and -1 represent the forward and backward one frame error in trajectory correspondences, and so on. In other words, the horizontal axis is the error, number of frames, between correspondent frames, and the vertical axis is a total number of frames that have a certain error. There are total 183 points in the sequences. Rank based approaches are not affected by this small disturbance, however, the fundamental matrix based approach degraded dramatically. We used the toolbox provided by Torr to compute the fundamental matrix and applied the linear ant non-linear approaches. We can conclude that the rank constraint based approach is much more stable than the fundamental matrix based approach.

3.2. Zoomed and Non-overlapping Sequences

In [2], Caspi and Irani propose an attractive method to align two non-overlapping video sequences. Their approach is based on the computation of inter-frame transformations along each video sequence. This approach requires two fixed cameras installed on a common platform. In their experiments, the scene is static, but the video cameras are moving. It is equivalent to the static cameras capturing the dynamic scene. Although the fields of views are non-overlapping, the spatial relationship (epipolar geometry) is still maintained.

We applied our method to sequences used in experiments of Caspi and Irani [2]. The first experiment contains one sequence captured by a camera with a wide FOV and the other captured by a camera with a zoomed FOV. The length of sequences is 300 frames. Fig. 4 shows the input frames. We tracked the lower left corner of the blue logo in both sequences to obtain trajectories. After alignment only nine frames had incorrect correspondences. Fig. 5 shows the results and the the histogram of matching error.

In the second experiment they used videos captured by the moving cameras. Fig. 6 shows the input sequences from the left and the right cameras. There are 80 frames in each video. We tracked the right-upper corner of the gate in the right camera sequence and the left-upper corner of the gate in the left camera sequence. The view-invariant DTW discovered 71 correct correspondences, and eight frames with one frame shift. Fig. 7 shows results of the trajectories and the histogram of matching error.

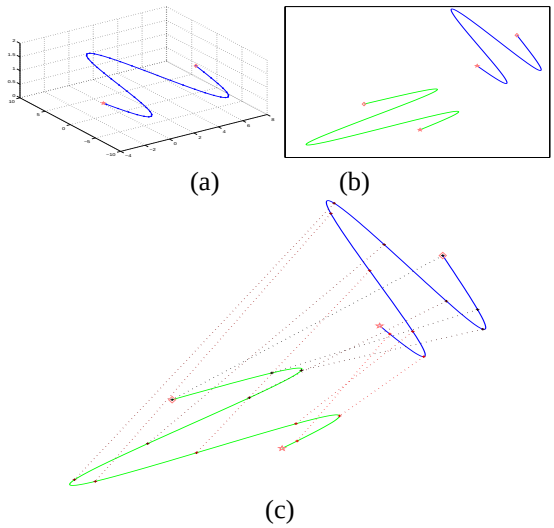


Figure 2: (a) A synthetic trajectory in 3D space. (b) The two projected trajectories of (a) in 2D space. (b) The view-invariant dynamic time warping result, where the dot lines connect the corresponding points

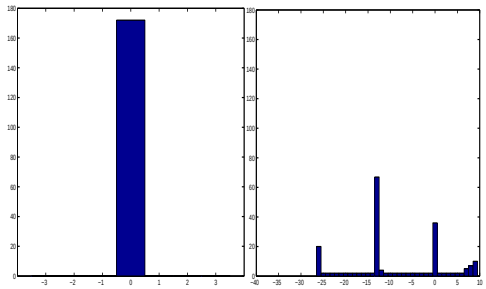


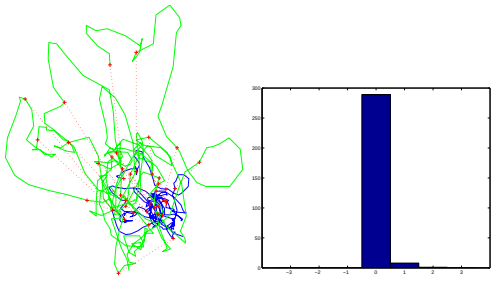
Figure 3: (a) The histogram of matching error using the rank constraint of the perspective camera with/without noise. (b) The histogram of matching error using fundamental matrix with very small noise in the data.

In the third experiment they used non-overlapping sequences. The first half of the videos contains the building around the football stadium. We tracked one feature on the wall of the football stadium and the corner of the window. Fig. 8 shows the input images, and Fig. 9 shows the results. The view-invariant DTW discovered 151 correct correspondences, 21 frames with one frame shift, and 28 frames with two frames shift. Fig. 8 shows the results of trajectories and the histogram of matching error. The error may due to the tracking error.

3.3. Syntheses of New Videos Containing Human Activities

From the previous experiments it is hard to evaluate the effectiveness of DTW function. Video sequences were captured simultaneously so the trajectories do not contain the dynamic change among the corresponding frames. Therefore, at different time and from distinct viewpoints we recorded videos people

Figure 4: The input sequences from Caspi and Irani's paper(frame 1,100,200,299), the first row is a wide field of view scene, and the second row is the zoomed scene.



(a) (b)

Figure 5: The correspondence result for the zoom sequences.

performing several identical actions.

In the first experiment, two students were moving their hands up and down with the different speeds. We recorded three videos using one camera. The first two videos were captured using static camera from the different viewpoints, while the third one was captured using a moving camera. The hands were tracked using the mean-shift tracker. We stabilize the frames of the third video (which was captured by moving camera) by subtracting the image coordinates of a static point (the corner of desk) from the image coordinates of the hand. There was a time-shift of approximately the half of the cycle in one of the videos relative to the other. We used the perspective camera model in the rank constraint approach to synchronize these videos. Despite of the change in the viewpoints and the non-linear time shift, our method successfully established the correspondence between videos. Fig. 10 shows the input videos. Fig. 11 shows the results of the view-invariant DTW. The results are quite impressive, since the large temporal variance had been compensated.

The next experiment dealt with synchronizing of videos that contain more complicated human activities. We recorded three dancers performed the same dance. For each dancer we captured two video sequences from two significant distinct view points. Fig. 12 shows the trajectories of the left feet of dancers in the six sequences. The difference between trajectories includes viewpoint difference, temporal difference and the difference due to the non-rigid motion of the dancers. We computed the temporal correspondence for each trajectory point with respect to the points in the other five trajectories. So there are total ($C_6^2 = 15$) computations. Based on the pair-wise correspondence we generated a video containing all six synchronized dance sequences, such that the sequence $\#1$ is warped towards the $\#6$ based on the warping function computed from the trajectories $\#1$ and $\#6$, the sequence $\#2$ is warped towards the sequence $\#6$ also,

Figure 6: The non-overlapping sequences (jump sequence), frame 1,27,54 and 80 are shown. The first row is from the left camera and the second row is from the right camera.

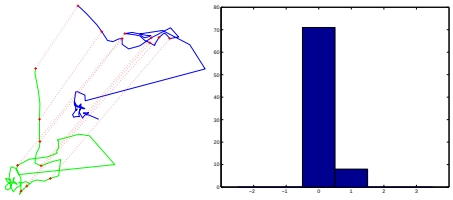


Figure 7: The view-invariant DTW correspondence result for jump sequences.

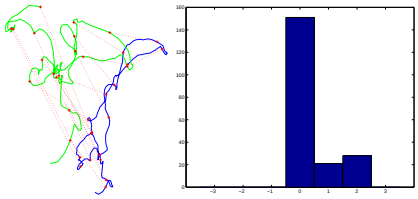
and so on. From example readers can notice from trajectories $\#3$ and $\#4$ that there is a huge spatial difference between trajectories. Fig. 13 shows one of the warping results, in which all sequences are warped toward the sequence $\#6$. Each row contains some key frames in the video, and the corresponding frames are shown in each column. Please reference to the supplemental materials to get the full size input/output movies. Although the videos contain large amount of non-rigid motion, our algorithm successfully computed the correspondence from one frame to the frames in other sequences. We are very happy to see that the algorithm runs very robustly and the results are synchronized with a high accuracy.

3.4. Computer Aid Training

The time-warping function is a path that minimizes the alignment error at each step through the similarity measure **E**. Each point from the path represents the correspondence between the i^{th} point in trajectory I and the j^{th} point in trajectory J. If many points in the trajectory I correspond to the same point in the trajectory J, then it means that the movement of sequence I is slower than the movement of sequence J at that moment. This observation gave us a clue for the performance estimation. We took sequences, $\#6$ as a model and $\#1$ as a test, and computed the warping path between them. Fig. 14(a) shows the result. From this figure we can notice that the dancer $\#1$ had a pause at around the frame 150. Fig. 14(b) shows the time-warping path between sequences $\#2$ and $\#6$. This figure shows the dancer $\#2$ did not decrease the speed at the frame 80. By this way, the users can find easily the places for improvement.

Fig. 15(a) shows the similarity measurement along the time-warping path for sequences $\#1$ and $\#6$. We noticed that the dancer did well overall, but she had a bad movement from frames 150 to 200. We checked the input sequence, and found that she lowered her leg from the upper most position around that time. Therefore, we concluded that she may need to improve that part. Fig.15(b) shows the similarity measurement for

Figure 8: The non-overlapping sequences (football sequence), frames 0,49,99 and 149 are shown. The first row is from the left camera and the second row is from the right camera. There are total over 300 frames in each sequence



(a) (b)

Figure 9: The view-invariant DTW correspondence result for football sequences. (a) shows the two trajectories and the corresponding points connected with dotted lines. (b) The histogram of matching error.

sequences #2 and #6, we detected the dancer #2 had the same problem as the dancer #1.

With the help of view-invariant DTW, we can easily develop a self-training system, such that the users (dancers #1 and #2) record their performance, and compare to the master's (dancer #3). Then the system will give suggestions about the speed and the extent of their movement. Note that the beginner's and master's camera viewpoints can be different. Therefore, this method has a great potential.

Figure 10: The human activity sequences. The first, second and third rows respectively shows the first, second and third input sequences, which are not synchronized. The columns are ordered as frame 0,20,40,60,80,100, and 120 for each sequence.

Figure 11: The output of the view invariant dynamic time warping. The columns represent the synchronized corresponding frames. Every 40th of the output frames are shown, they are 11,51,91,131,171,211,251,291.

References

[1] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *CVPR00*, pages 682–689, 2000.

[2] Yaron Caspi and Michal Irani. Alignment of Non-Overlapping sequences. In *ICCV'01*, pages 76–83, 2001.

[3] Yaron Caspi, Denis Simakov, and Michal Irani. Feature-based sequence-to-sequence matching. In *VAMODS (Vision and Modelling of Dynamic Scenes) workshop with ECCV*, Copenhagen, 2002.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, 2000.

[5] Trevor J. Darrell, Irfan A. Essa, and Alex P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Trans. PAMI*, 1995.

[6] David Demirdjian, Andrew Zisserman, and Radu Horaud. Stereo autocalibration from one plane. In *ECCV*, pages 625–639, 2000.

[7] M. Giese and T. Poggio. Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. In *Proceedings of the MVIEW 99 Symposium at CVPR*, pages 73–80, Fort Collins, CO, 1999.

[8] Radu Horaud and Gabriella Csurka. Autocalibration and euclidean reconstruction using rigid motion of a stereo rig. In *Proc. of the Sixth International Conference of Computer Vision*, pages 96–103, Bombay, India, 1998.

[9] V. Parameswaran and R. Chellappa. Quasi-invariants for human action representation and recognition. In *16th International Conference on Pattern Recognition*, volume 1, pages 307–310, 2002.

[10] S. M. Seitz and C. R. Dyer. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25:1–25, 1997.

[11] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *DARPA IU Workshop*, pages 521–527, 1998.

[12] the author is blanked for review. blanked for review. In *IEEE Workshop on Detection and Recognition of Events in Video (EVENT'01)*, Vancouver, Canada, July 2001.

[13] P. H. S. Torr and A Zisserman. Feature based methods for structure and motion estimation. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *International Workshop on Vision Algorithms*, pages 278–295, 1999.

[14] L. Wolf and A. Zomet. Sequence to sequence self-calibration. In *Proceedings of the European Conference on Computer Vision(ECCV)*, Copenhagen, May 2002.

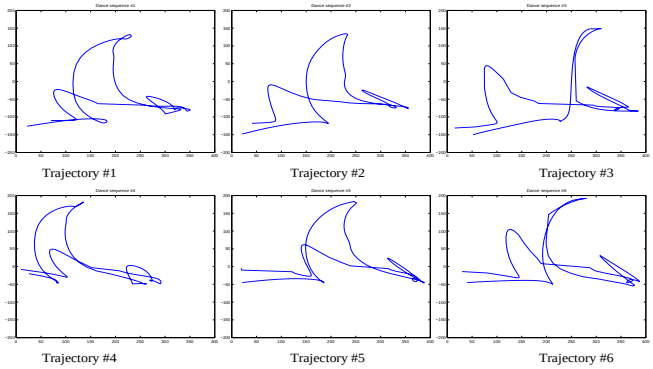
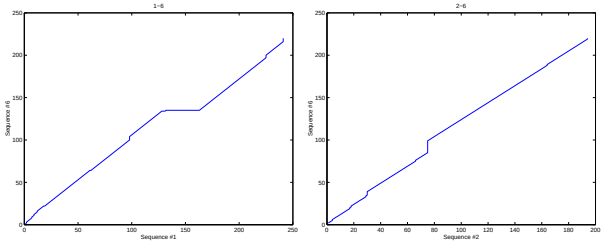


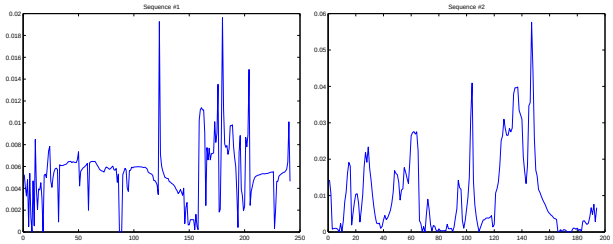
Figure 12: The trajectories of the right feet of dancers in 6 sequences. The first row contains trajectories $\#1, \#2$ and $\#3$ that correspond to the 1^{st}, 2^{nd} and 3^{rd} dancers respectively. The second row contains trajectories $\#4, \#5$ and $\#6$ that correspond to the 1^{st}, 2^{nd} and 3^{rd} dancers respectively also. Trajectories $\#1$

Figure 13: The key frames of the output sequences (the frame index is shown at the bottom of figures. The sequences $\#1, \#2, \#3, \#4, \#5$ are warped towards the sequence $\#6$ and are shown according to the rows. The 1^{st} and 4^{th} are correspond to the first dancer, the 2^{nd} and 5^{th}
correspond to the second dancer, and the 3^{rd} and 6^{th} correspond to the third dancer.



(a) (b)

Figure 14: (a) The time-warping path between the sequences $\#1$ and $\#6$, at the frame 150 there is a pause in sequence $\#1$. (b) The time-warping path between the sequences $\#2$ and $\#6$, at the frame 80 the sequence $\#2$ is faster.



(a) (b)

Figure 15: (a) The similarity measurement between the sequences $\#1$ and $\#6$, from frame 150 to 200 are contain large spatial difference. (b) The similarity measurement between the sequences $\#2$ and $\#6$, from frame 120 to 160 are contain large spatial difference.