AN EFFICIENT TECHNIQUE FOR MOTION-BASED VIEW-VARIANT VIDEO SEQUENCES SYNCHRONIZATION

C. Lu and M. Mandal

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
lcheng4@ualberta.ca and mandal@ece.ualberta.ca

ABSTRACT

In this paper, a novel technique is proposed for temporal alignment of video sequences with similar motions acquired using uncalibrated cameras. In this technique, we model the motion-based video temporal alignment problem as a spatial-temporal discrete trajectory point sets alignment problem. The trajectory of the interested object is tracked though out the videos. A probabilistic method is then developed to calculate the spatial correspondence between trajectory point sets. Next, the dynamic time warping technique (DTW) is applied to the spatial correspondence information to compute the temporal alignment of the videos. The experimental results show that the proposed technique provides a superior performance over existing techniques for videos with planar motion.

Index Terms —video alignment, video synchronization, temporal registration, dynamic time warping

1. INTRODUCTION

Alignment of video sequences plays a crucial role in applications such as super-resolution imaging [4], 3D visualization [13] and robust multi-view surveillance [8]. Past literature in video alignment or temporal registration can be broadly classified into two categories: video sequences of the same scene or video sequences of similar scenes. Several techniques have been proposed for synchronizing videos of the same scene. Lee *et al*. [8] proposed a videos synchronization technique, in which they estimated a time shift between two sequences based on alignment of centroids of moving objects in planar scene. Caspi *et al.* [4] calculated the spatial and temporal relationship between two sequences by minimizing the SSD error over extracted trajectories that are visible in both sequences. Padua *et al.* [10] extended [4] to align sequences based on scene points that need to be visible only in two consecutive frames.

For synchronization of videos of related scenes, Rao *et al.* [12] used rank constraint as the distance measure in a dynamic time warping (DTW) technique to align human activity-based videos. This is the first work reported in the literature that can deal with video sequences of correlated activities. Singh et al.[14] formulated a symmetric transfer error (STE) as a functional of regularized temporal warp. The technique determines the time warp that has smallest STE. Lu et al. [7] extended Singh's technique, using unbiased bidirectional dynamic time warping (UBDTW), to calculate the optimal warp for the alignment. We refer the Rao's, Singh's and Lu's technique as RBC, STE and UBD technique in this paper for simplicity.

Point sets alignment is an important technique for solving computer vision problems. Beal and McKay [2] proposed the iterative closest point (ICP) technique for 3D shape registration. The ICP technique computes the correspondences between two point sets based on distance criterion. Other techniques were developed which use soft-assignment of correspondences between two point sets instead of the binary assignment in ICP technique [5] [9]. Such probabilistic techniques perform better than the ICP technique in the case where noise and outliers are present. Point sets alignment technique has been used successfully in stereo matching, shape recognition and image registration. Although it has never been used in video synchronization technique, it has the potential to achieve good synchronization performance.

In this paper, we propose a novel technique for motion-based videos synchronization. We focus on the common case that the interested objects move over an approximated planar surface or when the distance between the cameras' centers of projection is small with respect to the scene motion. The main contribution of this work lies in formulating the motion-based video temporal alignment problem as a spatial-temporal discrete trajectory point sets alignment problem. The advantages of the proposed technique are: (1) it does not require overlapping views between videos to select corresponding feature points in the videos, i.e. it can be applied in videos containing different scenes; (2) it is able to deal with situations where videos containing complex dynamic object motion or noisy feature trajectory with consistent performance.

The rest of this paper is organized as follows. Section 2 presents review of related works in video synchronization. The proposed technique is presented in Section III. Performance evaluation of the proposed technique is presented in Section IV, followed by the conclusion.

978-1-61284-350-6/11/$26.00 ©2011 IEEE

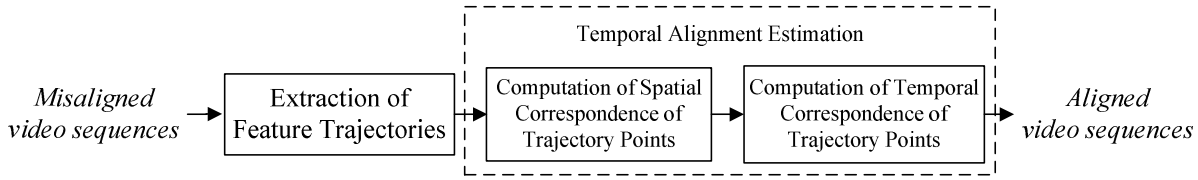

Fig. 3. Schematic of the proposed technique.

2. REVIEW OF RELATED WORK

Most video alignment techniques deal with video sequences of the same scene and hence assume that the temporal variant between videos is considered to be a linear time constraint [4] [10], such as $t' = s \cdot t + \Delta t$, where s is the ratio of frame rates and Δt is a fixed translational offset. However, for applications such as video search, video comparison and human activity recognition [11], we need to align video sequences from two different scenes.

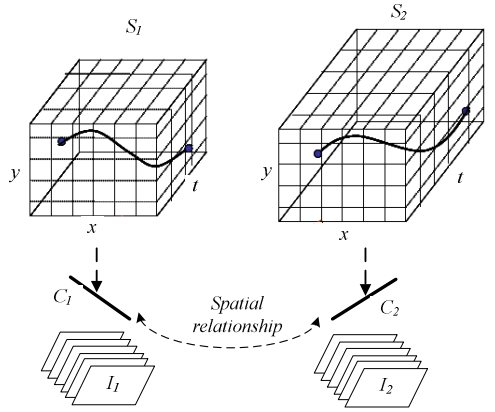

Fig. 1. Illustration of two different scenes acquired using two distinct cameras.

Assume that two cameras C_1 and C_2 view two independent scenes of similar activities, as shown in Fig.1. In Fig.1, C_1 (Camera 1) views 3D scene S_1 in View 1 and acquires video I_1. Similarly, C_2 (Camera 1) views another 3D scene S_2 in View 2 and acquires video I_2. Note that the motions in these two scenes are similar but have *dynamic time shift*, i.e. the linear time shift constraint in most of existing techniques (e.g. $t' = s \cdot t + \Delta t$) would not hold anymore.

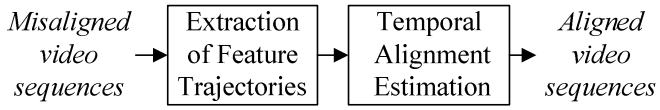

Fig. 2. Typical schematic of similar scene videos synchronization.

A typical schematic for calculating temporal alignment of similar scene videos is shown in Fig.2. The RBC, STE and UBD technique fall within this schematic. Note that for the sake of correlating two video sequences and representing the motion between the video sequences, features are extracted and tracked from two video sequences. Robust view-invariance tracker are used to generate feature trajectory. On their own, the feature trajectories are discrete representations of the motion in the scene. The variations of RBC, STE and UBD techniques lie in how to compute the alignment using DTW. RBC technique uses the rank constraint of corresponding points in two videos to measure the similarity. The similarity measurement is then used as the distance measure in DTW to calculate the dynamic time alignment function. Though the RBC technique does lead to good alignment of dynamic time-varying videos, the author of [12] note that if feature points are close to each other, their rank constraint will result in erroneous matching.

The STE technique projects the trajectory in one view into the other view, and then calculates the symmetric alignment using Euclidean distance based DTW. The technique determines the time warp that has the smallest STE as the final alignment. The UBD technique utilizes the symmetric alignment as the global constraint, and calculates the optimal warp using DTW.

The RBC, STE and UBD techniques suffer from two common drawbacks: (1) they require overlapping views between the first frame of videos in order to specify enough corresponding feature points; (2) imperfection of spatial relationship estimation, or noise in feature point tracking in the videos lead to the error in alignment.

3. PROPOSED TECHNIQUE

In this section, we present the proposed technique in details. The schematic of the proposed technique is shown in Fig.3. The technique consists of two main modules: extraction of feature trajectories and temporal alignment estimation. The temporal alignment estimation module has two steps: computation of the spatial correspondences of the trajectory points and computation of the temporal correspondences of the trajectory points. The spatio-temporal correspondence information is actually the synchronization information between the videos. The proposed temporal alignment estimation technique is presented in details in the following sections.

3.1. Extraction of feature trajectory

This module calculates the trajectory of a feature in the video. Let $F_1(x, y, t)$ and $F_2(x, y, t)$ denote the trajectories obtained from video I_1 and I_2, respectively, where (*x, y*) represent the feature coordinates at time *t*. Similar to the existing technique, robust tracker, e.g. KLT or mean shift [3], can be used to generate feature trajectory. Denote the point sets in trajectory F_1 and F_2 as X and Y, respectively, which are defined as follows.

$$X = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_N^T \end{bmatrix} = \begin{bmatrix} F_1(x_1) & F_1(y_1) & 1 \\ F_1(x_2) & F_1(y_2) & 1 \\ \vdots & \vdots & \vdots \\ F_1(x_N) & F_1(y_N) & 1 \end{bmatrix}_{N\times 3}$$

$$Y = \begin{bmatrix} Y_1^T \\ Y_2^T \\ \vdots \\ Y_M^T \end{bmatrix} = \begin{bmatrix} F_2(x_1) & F_2(y_1) & 1 \\ F_2(x_2) & F_2(y_2) & 1 \\ \vdots & \vdots & \vdots \\ F_2(x_M) & F_2(y_M) & 1 \end{bmatrix}_{M\times 3} \quad (1)$$

Note that N and M are the total number of points in trajectory F_1 and F_2, respectively. $F(x_n)$ and $F(y_n)$ represent the coordinates of the *nth* trajectory point. Each discrete point in trajectory is written in homogenous form. The indices of the point also indicate the temporal information, i.e. the number of frames in two videos.

3.2. Compute spatial correspondence of trajectory points

In this section, we aim to calculate the spatial correspondence of trajectory points. The estimated spatial point correspondence is then used for computation of the temporal correspondence in the next section.

We model the spatial transformation of the two trajectory point sets by homography matrix. The homography is a 3x3 matrix containing 9 unknowns $h_1, h_2, \ldots, h_9$. Note that one pair of correspondence points (X_n, Y_m) in two different views satisfies the following equation.

$$X_n = \begin{bmatrix} F_1(x_n) \\ F_1(y_n) \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_8 & h_7 & h_9 \end{bmatrix} \begin{bmatrix} F_2(x_m) \\ F_2(y_m) \\ 1 \end{bmatrix} = HY_m \quad (2)$$

In order to calculate the spatial correspondences, we treat Y as the centroids of a Gaussian mixture model (GMM) [1], and assume X as the data points generated by the GMM independently. We then re-parameterize the GMM centroids Y to the data points X by maximizing the log likelihood function. The probability of the m^{th} Gaussian component in Y generating data point X_n, i.e. the posterior probability $P(Y_m|X_n)$, is the correspondence we are looking for. The idea for computation of spatial correspondences is similar to the general framework proposed by Myronenko et. al. [9], but we model the spatial relationship with the homography. The conditional Gaussian probability density function is defined as follows.

$$p(X_n \mid Y_m, \phi) = \frac{1}{(2\pi\sigma^2)^{3/2}} e^{\left(-\frac{\| X_n - HY_m \|^2}{2\sigma^2}\right)} \quad (3)$$

where ϕ represents the parameters set H and σ^2. In this paper we assume that the GMM components have identical covariance σ^2 and membership probabilities. The GMM probability density function can then be expressed as follows.

$$p(X_n \mid \phi) = \sum_{m=1}^{M} P(Y_m) p(X_n \mid Y_m, \phi) = \sum_{m=1}^{M} \frac{1}{M} p(X_n \mid Y_m, \phi) \quad (4)$$

The log-likelihood function of the GMM probability density function can be calculated as follows.

$$L(\phi) = \log \prod_{n=1}^{N} p(X_n \mid \phi) = \sum_{n=1}^{N} \log \sum_{m=1}^{M} \frac{1}{M} p(X_n \mid Y_m, \phi) \quad (5)$$

3.2.1. The EM algorithm

Note that finding the maximum likelihood with respect to the parameters ϕ using Eq. (5) is difficult, since we cannot find a closed form solution for it. The Expectation Maximization (EM) algorithm [1] is therefore used to estimate the parameters. The EM algorithm focuses on finding the maximum log-likelihood solutions posed in terms of missing data. In our case, the missing data is actually the point-to-centroid correspondence $P(Y_m|X_n)$, i.e. the posterior of the GMM centroid Y_m given data X_n. By introducing the posterior $P(Y_m|X_n)$, the EM algorithm estimates the parameters in an iterative framework. The revised log-likelihood function of the GMM probability density function in the EM algorithm is defined as follows [1]:

$$L(\phi^{(i)}) = \sum_{n=1}^{N} \sum_{m=1}^{M} P(Y_m \mid X_n, \phi^{(i-1)}) \log\left(\frac{1}{M} p(X_n \mid Y_m, \phi^{(i)})\right) \quad (6)$$

The EM algorithm estimates $P(Y_m|X_n)$ and ϕ iteratively in a two-steps manner. In the first step (E-step), the posterior $P(Y_m|X_n, \phi^{(i-1)})$ are estimated using Bayes rule as follows.

$$P(Y_m \mid X_n, \phi^{(i-1)}) = \frac{\frac{1}{M} p(X_n \mid Y_m, \phi^{(i-1)})}{p(X_n \mid \phi^{(i-1)})} = \frac{e^{-\frac{1}{2}\frac{\| X_n - H^{(i-1)} Y_m \|^2}{\sigma^{2(i-1)}}}}{\sum_{m=1}^{M} e^{-\frac{1}{2}\frac{\| X_n - H^{(i-1)} Y_m \|^2}{\sigma^{2(i-1)}}}} \quad (7)$$

where $\phi^{(i-1)}$ is the parameters estimated at (i-1)*th* iteration. Note that to avoid ambiguity, we denote the variable with superscript within parenthesis as the *ith* version of a variable. For example, $\sigma^{2(i)}$ represents the *ith* version of σ^2, $H^{(i)}$ represents the *ith* version of H in the iteration framework.

In the second step (M-step), we estimate the $\phi^{(i)}$ by maximizing the $L(\phi^{(i)})$ in Eq. (6). The EM algorithm iterates these two steps until converges. The final version of estimated posterior probability $P(Y_m|X_n)$ is the spatial correspondences of two point sets we are interested in. Substituting Eq. (3) into Eq. (6) and ignoring the constant terms, we have the log-likelihood function $L(\phi^{(i)} \mid \phi^{(i-1)})$ in M-step as follows.

$$\begin{aligned} L(\phi^{(i)} \mid \phi^{(i-1)}) = &-\sum_{n=1}^{N} \sum_{m=1}^{M} P(Y_m \mid X_n, \phi^{(i-1)}) \frac{\| X_n - H^{(i)} Y_m \|^2}{2\sigma^{2(i)}} \\ &- \frac{3}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} P(Y_m \mid X_n, \phi^{(i-1)}) \log \sigma^{2(i)} \end{aligned} \quad (8)$$

It can be shown that the solution for maximizing Eq. (8) at *ith* M-step with respect to $H^{(i)}$ and $\sigma^{2(i)}$ is as follows:

$$H^{(i)} = (X^T P_{MN}^{(i-1)T} Y)(Y^T d(P_{MN}^{(i-1)}\mathbf{1}) Y)^{-1} \quad (9)$$

$$\sigma^{2(i)} = \frac{Tr(X^T d(P^{(i-1)T_{MN}}\mathbf{1})X) - Tr(X^T P^{(i-1)T_{MN}} Y H^{(i)T})}{3\sum_{n=1}^{N} \sum_{m=1}^{M} P(Y_m \mid X_n, \phi^{(i-1)})} \quad (10)$$

where $\mathbf{1}$ is the column vector of all ones, $d(P_{MN}^{(i-1)}\mathbf{1})$ is the diagonal matrix formed from the vector $P_{MN}^{(i-1)}\mathbf{1}$, *Tr* is the trace operation for a matrix, The spatial point correspondence matrix P_{MN} is given by

$$P_{MN} = \begin{bmatrix} P(Y_1 \mid X_1) & P(Y_1 \mid X_2) & \dots & P(Y_1 \mid X_N) \\ P(Y_2 \mid X_1) & P(Y_2 \mid X_2) & \dots & P(Y_2 \mid X_N) \\ \vdots & \vdots & \vdots & \vdots \\ P(Y_M \mid X_1) & P(Y_M \mid X_2) & \dots & P(Y_M \mid X_N) \end{bmatrix}_{M \times N} \quad (11)$$

Note that each element in P_{MN} is computed using Eq. (7). The EM algorithm for calculating the point-to-centroid correspondence in our problem is summarized in Table 1.

Table 1. Pseudo code for calculation of spatial point-to-centroid correspondence using EM algorithm

Input: point set X and Y (see Eq.(1))
Initialization:
$H^{(0)} = I_{3x3}$, $\sigma^{2(0)} = \frac{1}{3NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\|X_n - H^{(0)}Y_m\|^2$
For *ith* iteration
E-step:
Compute $P(Y_m \mid X_n, \phi^{(i-1)})$ using Eq. (7);
Compute $P_{MN}^{(i-1)}$ using Eq. (11)
M-step:
Compute $H^{(i)}$ using Eq. (9)
Compute $\sigma^{2(i)}$ using Eq. (10)
Until converge (i.e. $E(\phi^{(i)} \mid \phi^{(i-1)}) \le E(\phi^{(i-1)} \mid \phi^{(i-2)})$**)**
Output: Final version of P_{MN}

3.3. Compute Temporal Correspondence of Trajectory Points

In the previous section, the spatial correspondence module estimated spatial point correspondence matrix P_{MN}. In this section, we present the procedure to calculate the exact correspondence using P_{MN}.

In a general point set alignment problem, for each point X_n in X, the exact corresponding points can be determined by choosing the mth point Y_m in Y such that the posterior is the maximum, i.e.

$$m^* = \arg\max_m (P(Y_1 \mid X_n), P(Y_2 \mid X_n), \dots, P(Y_m \mid X_n))$$

However, in our video synchronization problem, it may incur errors in matching if the feature trajectory itself has intersections at different time instances or the trajectory is noisy. Because what we did in section 3.2 is to estimate the posterior based on the spatial information. In order to solve the problem stated above, we propose to utilize the temporal constraint on the trajectory points and compute the actual correspondences using the DTW technique based on the obtained spatial point correspondence matrix P_{MN}. Denote the temporal alignment between two trajectories as the warp. We construct the warp $\mathcal{W}$ as follows:

$$\mathcal{W} = w_1, w_2, ..., w_K$$
$$max(N, M) \le K < N + M$$

where N and M are the length of trajectories F_1 and F_2 obtained from video I_1 and I_2, respectively. The k^{th} element of the warp $\mathcal{W}$ is $w_k = (m, n)$. The warp satisfies the boundary conditions, continuity conditions and monotonicity conditions which are explained in [15]. The traditional DTW technique computes the warping based on the Euclidean distance between two sequences and choose the warp which has the minimum accumulated distance as the optimal warp. The proposed technique computes the warp based on the probabilistic value, and chooses the warp which has the maximum accumulated probability as the optimal warp. The accumulated probability of a warp is defined as follows:

$$AccPr(\mathcal{W}) = \sum_{k=1}^{K} P(w_k)$$

where $P(w_k) = P(Y_m|X_n)$ is the posterior value of the given indices (m, n) in the k^{th} element of the warp.

In order to find the optimal warp, a M x N accumulated distance matrix is created. The element in the accumulated probability matrix is calculated as follows.

$$\mathcal{P}(m,n) = P(Y_m|X_n) + \max[\mathcal{P}(m-1,n), \mathcal{P}(m-1,n-1), \mathcal{P}(m,n-1)]$$

A greedy search technique is then employed in the accumulated probability matrix to find the optimal warp $\mathcal{W}$ such that the $AccPr(\mathcal{W})$ is maximum.

4. EXPERIMENT AND COMPARATIVE ANALYSIS

In this section, we evaluated the proposed technique using both synthetic trajectories and real image sequences. We also implemented the RCB technique, STE technique and UBD technique as described in [12], [14], [7] that deal with aligning videos of similar motions. Our test cases and results are presented next. All experiments were run on a 2.4 GHz Intel Core II Duo CPU with 3GB RAM using MATLAB 7.04.

4.1. Synthetic data evaluation

For synthetic data evaluation, we generate a 100 frames long complex planar trajectory, using a pseudo-random number generator to simulate the motion in videos. The trajectory is then projected onto two image planes using user defined camera projection matrices. The camera matrices are designed so that the acquisition emulates a homography, and are used only for generation purpose and not thereafter. In order to simulate the dynamic time shift, a 60 frames long time warp is then applied to a section of one of the trajectory projections. We then refer this trajectory as time-warped trajectory and its length is 160 frames long, which is different from the planar trajectory. The proposed technique, the RCB and the STE techniques are then applied to the synthetic trajectories to compute the alignment between them. This process is repeated on 50 different synthetic trajectories.

We also evaluated the effect of noisy trajectories on the proposed technique. Normally distributed and zero mean noise with various values of variance (σ^2) was added to the synthetic feature trajectories. A synthetic noisy trajectory example (with $\sigma^2 = 0.1$) is shown in Fig.4. Fig.4. (a) and (b) are the noisy projected trajectories on two image planes, respectively. Note that the red dots area in Fig.4 (b) is the warped part which simulates the dynamic time shift between two similar motions. Fig.4 (c) shows the alignment results obtained by the RCB, STE, and UBD proposed technique. The

x-axis and y-axis in Fig.4 (c) represent the indices of time-warped trajectory and trajectory in another image plane, respectively. The ground truth alignment is shown as solid red line. It is note that the RCB, STE and UBD technique behave badly for such complex and large view-variant trajectories.

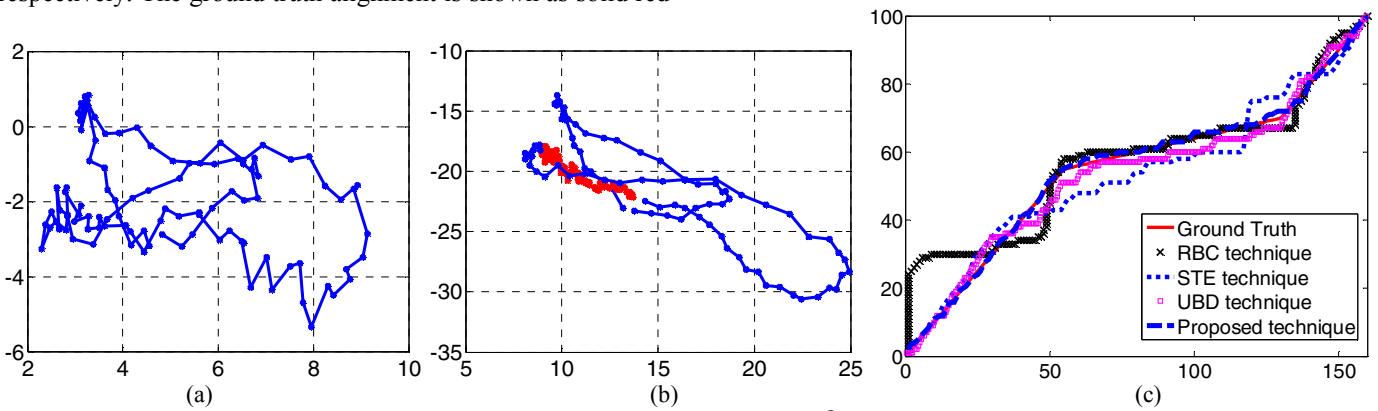

Fig. 4. An example of alignments for view-variance noisy synthetic trajectory (with $\sigma^2 = 0.1$): (a) Noisy projected trajectory in one image plane (b) The noisy projected time-warped trajectory in another image plane (c) The comparison of the alignment results.

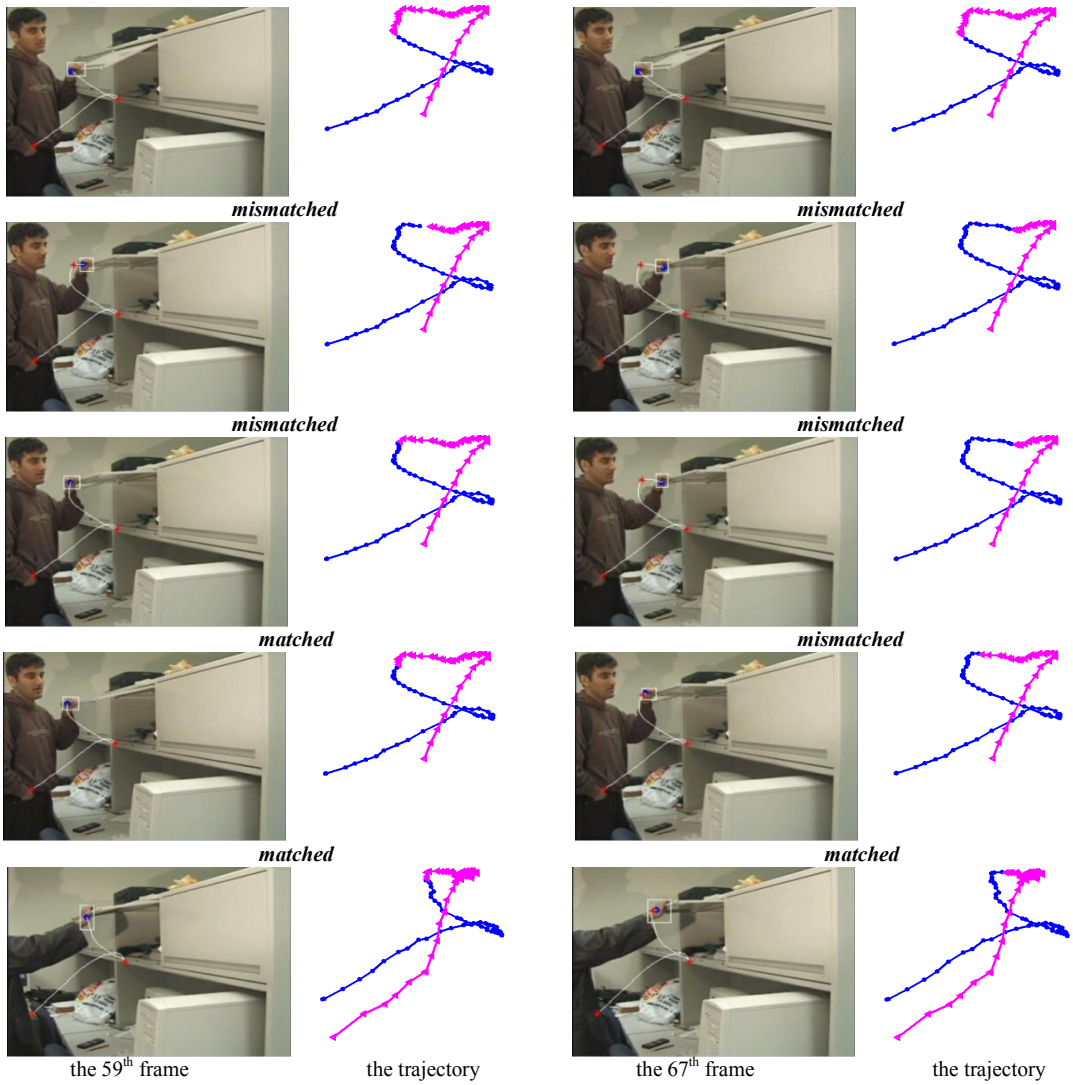

Fig. 5. Visual comparisons of the alignments computed using the RCB technique (the first row), the STE technique (the second row), and the UBD technique (the third row) and the proposed technique (the fourth row) on UCF1 video sequence. The fifth row shows the 59th, 67th frames of video UCF2 and the trajectories. The first to fourth rows shows the aligned frame obtained using the RCB, the STE, the UBD technique and the proposed technique.

The results of alignment of smooth and noisy trajectories with both the RCB and the proposed techniques are shown in Table 2, where the mean absolute differences between the actual and computed frame correspondence is reported as the alignment error. It is important to note that the proposed technique provides consistent performance under the condition of noisy trajectory and outperforms the other techniques significantly in the synthetic experiment.

Table 2. Alignment errors of RCB, STE, UBD and proposed technique for synthetic data

Noise σ^2	RCB	STE	UBD	Proposed Technique
0	4.73	2.70	2.18	0.56
0.1	5.69	3.79	2.30	0.82
1	7.27	3.91	2.97	1.74

Table 3. Alignment computed using RCB, STE, UBD and proposed technique for real video

Frame No. in UCF1	RCB	STE	UBD	Proposed Technique
59	36	51	46	42
67	36	56	55	49

4.2 Real data evaluation

For tests on real video sequences, we used video sequences, cab2-4.1.mpg and cab2-4.4.mpg, provided by Rao *et al.* at http://server.cs.ucf.edu/~vision/ (we refer these videos as UCF1 and UCF2 later). These videos recorded the activity of opening a drawer which is 84 and 174 frames long, respectively. Feature trajectories were available for the UCF video files by tracing the hand. Note that the ground truth information is not available. However, we can use visual judgment to check if the alignment is correct. Two representative frames, i.e. the 59^{th} and 67^{th} frames of video UCF2 is shown in the fifth row of Fig.5. The alignment computed for these two frames using the RCB technique, the STE technique, the UBD technique and the proposed technique are summarized in Table III and show in the first, the second, the third and the fourth row of Fig.5, respectively. Note that the alignment result can be compared using the tracked hand in the frame. In order to show the comparison more clear, we also present the motion trajectories beside the frames. The blue dots trajectory shows the motion trajectory up to the index we obtained. The magenta triangle trajectory represents the remaining motion. We can compare the first four rows, which are the corresponding frames computed using the RCB, STE, UBD and the proposed technique, to the last row that containing the 59^{th} and 67^{th} frames of video UCF2. If the computed aligned frame is matched, we will mark '*matched*' under the frame-trajectory pair, otherwise '*mismatched*'. It is shown that the proposed technique provides a relatively accurate temporal matching in this real video case.

5. CONCLUSION

In this paper, we proposed a novel efficient technique for motion-based videos synchronization. The proposed technique is able to synchronize videos containing complex motions with dynamic time shift in an efficient way. Comparative analysis with the RCB technique, STE technique and UBD technique demonstrated the significant advantage in video temporal alignment using our proposed technique. Although the number of videos to be synchronized is assume to be two in this paper. However, the proposed technique can deal with the problem where the number of videos is greater than two by setting one video as the reference and compute the temporal alignments of other videos with respect to the reference.

6. REFERENCES

[1] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1996.

[2] P. J. Besl and N. D. McKay, " A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, pp. 239-256, 1992.

[3] D. Comaniciu*, et al.*, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conference on Computer Vision and Pattern Recognition, Proceedings, Vol Ii,* pp. 142-149, 2000.

[4] Y. Caspi, and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 24, no. 11, pp. 1409-1424, Nov, 2002.

[5] S. Gold, C.P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness, "New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence," *Proc. Advances in Neural Information Processing Systems*, vol. 7, pp. 957-964, 1994.

[6] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press: London, 2004.

[7] C. Lu, M. Mandal. "Efficient Temporal Alignment of Video Sequences Using Unbiased Bidirectional Dynamic Time Warping," *Journal of Electronic Imaging*, in Press.

[8] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 758-767, Aug, 2000.

[9] A. Myronenko and X. B. Song, "Point Set Registration: Coherent Point Drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2262-2275, Dec 2010.

[10] F. L. C. Padua, R. L. Carceroni, G. Santos et al., "Linear Sequence-to-Sequence Alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 304-320, Feb, 2010.

[11] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203-226, Nov, 2002.

[12] C. Rao, A. Gritai, M. Shah and T. F. S. Mahmood, "View-invariant alignment and matching of video sequences," *In proc. ICCV03*, pp. 939-945, 2003.

[13] M. Singh, A. Basu, and M. Mandal, "Event dynamics based temporal registration," *IEEE Transactions on Multimedia,* vol. 9, no. 5, pp. 1004-1015, Aug, 2007.

[14] M. Singh, et al., "Optimization of Symmetric Transfer Error for Sub-frame Video Synchronization," *in Computer Vision - ECCV 2008, Pt Ii, Proceedings*. vol. 5303, D. Forsyth, et al., Eds., ed, 2008, pp. 554-567.

[15] H. Sakoe and S.Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol.26, no.1, pp.43-49, 1978.