IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

NOVEMBER 2017 VOLUME 27 NUMBER 11 ITCTEM (ISSN 1051-8215)

PAPERS

(Contents Continued on Back Cover)

(Contents Continued from Front Cover)

Motion-Based Temporal Alignment of Independently Moving Cameras

Xue Wang, Jianbo Shi, Hyun Soo Park, and Qing Wang, *Member, IEEE*

***Abstract*— This paper presents a method to establish a nonlinear temporal correspondence between two video sequences captured by cameras independently moving in a dynamic 3D scene. We assume that the 3D spatial poses of the cameras are known for each frame. With predefined trajectory basis, the coefficients of the reconstructed trajectory of a moving scene point reflect the rhythm in motion. A robust rank constraint from the coefficient matrices is exploited to measure the spatiotemporal alignment quality for every feasible pair of video fragments. Point correspondences across sequences are not required or even it is possible that different points are tracked in different sequences, only if they satisfy the assumption that every 3D point tracked in the observed sequence can be described as a linear combination of a subset of the 3D points tracked in the reference sequence. Synchronization is then performed using a graph-based search algorithm to find the globally optimal path that minimizes both spatial and temporal misalignments. Our algorithm can use both complete and incomplete feature trajectories along time, and is robust to mild outliers. We verify the robustness and performance of the proposed approach on synthetic data as well as on challenging real video sequences.**

***Index Terms*— Nonrigid structure from motion, rank constraint, trajectory basis, video synchronization.**

I. Introduction

IMAGINE a group of people wearing first-person cameras and performing an activity together (Fig. 1). These videos captured by independently moving cameras can be used together to obtain a complete 3D scene or recognize human action. For most of such computer vision applications, multiple unsynchronized video streams need to be synchronized at first. We expect that the synchronization could be achieved automatically using visual information only. This is a difficult task in practice.

Video synchronization is part of a more general video alignment problem that occurs in tasks such as human action recognition, video retrieval, multiview surveillance, and 3D visualization. Videos must be aligned both spatially and temporally. Spatial alignment computes the geometrical transformation of 2D or 3D coordinate systems of aligned frames; therefore, the object of interest is in correspondence. Temporal alignment computes 1D temporal mapping by aligning frames to achieve good spatial alignment. Most previous methods consider the problem in 2D by computing the best fitting geometry (e.g., homography, perspective, or affine projection model) and checking the residual errors. These methods, referred to as 2D analysis, usually require large overlap between synchronized frames, which is fulfilled only for narrow-baseline views. Alternatively, one can also consider the problem in 3D, for instance, by backprojecting the points as lines into the 3D world and looking for intersecting lines [1]. These methods, referred to as 3D analysis, require additional knowledge (e.g., camera poses) or assume certain projection models.

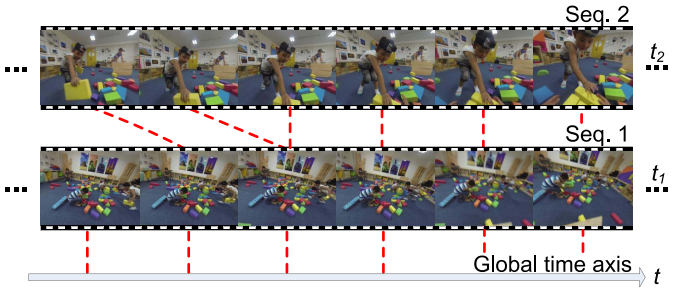


Fig. 1. Our algorithm synchronizes multiple independent video streams recorded in a social event. Large view variations, independent camera motions, and nonlinear temporal relationship make the task extremely challenging. The local time axis t_1 is referred to as the global time axis t in this example.

Jointly reasoning about spatial and temporal alignment improves the robustness of the system. There are two main challenges. First, explicit 2D or 3D spatial alignment is very difficult to compute for independently moving cameras on a dynamically changing scene with multiple moving objects. Second, due to nonpredictable frame drops, temporal context constraints (i.e., continuity) cannot be applied everywhere for temporal alignment.

This paper describes an algorithm that synchronizes two video sequences captured by free-moving cameras. The key insight is the spatiotemporal rhythm in the 3D motion of a human body. Both the geometrical configuration and the trajectories of body parts are strong cues for alignment, which are coupled in the rhythm. Our method uses sparse space–time feature trajectories as input and avoids the need

Manuscript received December 25, 2015; revised April 18, 2016; accepted June 11, 2016. Date of publication June 15, 2016; date of current version November 8, 2017. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61272287 and Grant 61531014. This paper was recommended by Associate Editor C. Zhang.

X. Wang and Q. Wang are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072 China (e-mail: xwang@mail.nwpu.edu.cn; qwang@nwpu.edu.cn).

J. Shi and H. S. Park are with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: jshi@seas.upenn.edu; hypar@seas.upenn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2581659

of correspondences across cameras. Moreover, the features tracked in different cameras could be different. Using the image observation, we measure the *feasibility* of spatiotemporal alignment between two video fragments using a rank constraint based on 3D trajectory reconstruction.

We do not put any restrictive constraints on the scene or on the camera motions, except that the cameras view one or more moving objects simultaneously. By realizing that the individual time delay is probably not an integer (i.e., the closest frame does not correspond exactly), instead of estimating the temporal mapping with subframe accuracy, we try to find the temporal closest frame. In addition, unlike the methods that assume that a sufficient amount of features can be tracked throughout both sequences, our approach utilizes both complete and incomplete feature trajectories along time and tolerates mild outliers.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the rank constraint derived from the coefficient vectors of 3D trajectory reconstruction. Section IV presents the robust optimal path search algorithm for nonlinear temporal alignment. Section V provides the experimental results for both synthetic and real data and discusses the robustness and performance of our algorithm. Finally, the main conclusions are drawn in Section VI.

II. Previous Work

Most related contributions assume stationary or rigidly fixed cameras [2]–[14]. Hence, a fixed spatial transformation between corresponding frames is guaranteed and need not be re-estimated at runtime. Once an event has been identified in two such videos, a temporal mapping between the sequences can be globally described by simple parametric models, like the constant offset model [5], [6], [9], [12] or 1D affine model [2]–[4], [7], [8], [10], [13]. Nonlinear temporal mapping is used to cope with free form of time correspondence [11], [14], e.g., nonpredictable frame drops, human action recognition, and video retrieval. Assuming simultaneous recording, this kind of temporal rigidity is preserved even for independently moving cameras [1], [15]–[18]. If related videos are captured at different points in time, for the video matching to be possible, [19]–[24] assume approximately coincident camera trajectories, which guarantee that corresponding frames are captured from similar viewpoints and have sufficient overlap in their field of views (FOVs).

Considering the input data of these methods, most fall into the feature-based category, which usually require complete features reliably tracked over the entire sequences and known correspondences across sequences. Such feature-based methods rely on the existence of a geometric entity that somehow constraints the relationship between the coordinate systems of two corresponding frames. Commonly exploited geometric constraints include plane-induced homography [3], [4], binocular epipolar geometry constraint [4], [12], [13], [16], deficient rank conditions arose from special projection models [5]–[7], [17], affine transformation [14], tri-focal tensor [15], feature movements [18], [20], matching image points [19], [23], and so on. The intensity-based synchronization methods are solely based on the image intensity [5], [6], [8], Fourier transform of image intensity [9], or dynamic texture [10]. They try to minimize the sum of squared differences between the sequences that can be spatially and temporally warped through a parametric model. Since all the pixels can provide constraints to such a model, feature tracking and matching can be avoided. The camera movement-based synchronization method [2] assumes that the two cameras are attached closely together and are moved jointly in space, and then the *consistent temporal behavior* can be used to recover the spatial and temporal transformations between two nonoverlapping sequences.

Our scenario is most closely related to the work in [1], [15], [17], and [18], which focuses on video synchronization for independently moving cameras and dynamic 3D scenes. With a scaled orthographic projection model assumption, Tuytelaars and Van Gool [1] evaluate the line-to-line distance of the backprojection 3D lines of the matching points to estimate a constant time offset between two sequences. Lei and Yang [15] use the tri-ocular geometric constraint of point/line features to build the timeline maps (integral time offsets) for multiple sequences to be synchronized. Tresadern and Reid [17] develop a unified rank constraint framework for homography, perspective, and affine projection models to recover a linear synchronization with subframe accuracy. These methods assume that the features are matched across sequences and tracked successfully throughout each sequence, which are difficult to obtain automatically in wide baseline conditions. Dexter *et al.* [18] adopt time-adaptive descriptors of image sequences from self-similarity matrices to perform nonlinear synchronization. They do not impose restrictive assumptions as complete feature trajectories along time or correspondences across sequences. However, they use static points in the background to estimate a dominant motion to compensate for modest camera motion, which works only for distant views or planar scenes.

The presented method is inspired by [5] and [6], which use rank constraint of a matrix of image measurements to define its *energy* above an expected rank bound. This *energy* is minimized when the structure is most consistent between synchronized sequences. The method requires complete feature trajectories but does not require exact correspondences across sequences. Instead, they make a weaker assumption that the points tracked in the second sequence could be expressed as a fixed linear combination of a subset of the points tracked in the reference sequence. A similar assumption is made in our method. The limitation is that it recovers the integral time offset by globally pooling data from groups of 2D point trajectories. As a result, their method works only for fixed affine cameras that have the same frame rate. In contrast, we propose a novel rank constraint using reconstructed 3D point trajectories to recover a nonlinear time warp for free-moving cameras.

III. Motion-Based Rank Constraint

Since the cameras undergo different motions in both sequences, the point trajectories of one moving 3D point captured by different cameras are dissimilar. To remove the effect of camera motion, we reconstruct the 3D point trajectory as a

linear combination of predefined basis trajectories [25], [26], such as the discrete cosine transform (DCT).

For a given tth (time stamp) camera projection matrix, $\mathbf{P}^{(t)} \in \mathbb{R}^{3\times 4}$, let a point in 3D, $\mathbf{X}^{(t)} = [X^{(t)}\ Y^{(t)}\ Z^{(t)}]^\top$, be imaged as $\mathbf{x}^{(t)} = [x^{(t)}\ y^{(t)}]^\top$. This projection is defined up to scale

$$\begin{bmatrix}\mathbf{x}^{(t)}\\ 1\end{bmatrix} \simeq \mathbf{P}^{(t)}\begin{bmatrix}\mathbf{X}^{(t)}\\ 1\end{bmatrix}, \quad \text{or} \quad \begin{bmatrix}\mathbf{x}^{(t)}\\ 1\end{bmatrix}_\times \mathbf{P}^{(t)}\begin{bmatrix}\mathbf{X}^{(t)}\\ 1\end{bmatrix} = \mathbf{0} \tag{1}$$

where $[\cdot]_\times$ is the skew symmetric representation of cross product. By taking F time stamps, a closed form for reconstructing the 3D trajectory of the point $\mathbf{X}$ can be formulated as [26]

$$\begin{bmatrix}\mathbf{Q}^{(1)} & & \\ & \ddots & \\ & & \mathbf{Q}^{(F)}\end{bmatrix}\begin{bmatrix}\mathbf{X}^{(1)}\\ \vdots\\ \mathbf{X}^{(F)}\end{bmatrix} = \begin{bmatrix}\mathbf{q}^{(1)}\\ \vdots\\ \mathbf{q}^{(F)}\end{bmatrix}, \quad \text{or} \quad \mathbf{QX} = \mathbf{q} \tag{2}$$

where $\mathbf{Q}^{(t)} = ([\vec{\mathbf{x}}^{(t)}]_\times \mathbf{P}^{(t)}_{1:3})_{1:2}$ and $\mathbf{q}^{(t)} = (-[\vec{\mathbf{x}}^{(t)}]_\times \mathbf{P}^{(t)}_4)_{1:2}$. Note $\vec{\mathbf{x}}^{(t)} = [x^{(t)}\ y^{(t)}\ 1]^\top$ is the homogeneous coordinates of $\mathbf{x}^{(t)}$. $\mathbf{P}^{(t)}_{1:3}$ and $\mathbf{P}^{(t)}_4$ are the matrices made of the first three columns and the last column of $\mathbf{P}^{(t)}$, respectively. $(\cdot)_{1:2}$ is the matrix made of the first two rows from $(\cdot)$.

The 3D point trajectory is approximated using a linear combination of the DCT basis, which can be described as

$$\mathbf{X} = [\mathbf{X}^{(1)\top}\ \ \ldots\ \ \mathbf{X}^{(F)\top}]^\top \approx \boldsymbol{\Theta}_1\beta_1 + \cdots + \boldsymbol{\Theta}_{3K}\beta_{3K} = \boldsymbol{\Theta}\boldsymbol{\beta} \tag{3}$$

where $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_1 \ldots \boldsymbol{\Theta}_{3K}] \in \mathbb{R}^{3F\times 3K}$ is the trajectory basis matrix, $\boldsymbol{\beta} = [\beta_1\ \ldots\ \beta_{3K}]^\top \in \mathbb{R}^{3K}$ is the coefficients of a point trajectory, and K is the number of bases per coordinate. By plugging (3) into (2), an overconstrained system can be derived by choosing K such that $2F \geq 3K$

$$\mathbf{Q}\boldsymbol{\Theta}\boldsymbol{\beta} = \mathbf{q}. \tag{4}$$

Equation (4) is a linear least square system for reconstructing a point trajectory, which is proved to be capable of providing an efficient, numerically stable, and globally optimal solution [26]. Fast and random camera motion results in high reconstructibility. If the same trajectory is seen by another moving camera (registered in the same world coordinate system with $\mathbf{P}_i$), we get

$$\hat{\mathbf{Q}}\mathbf{X} = \hat{\mathbf{Q}}\boldsymbol{\Theta}\hat{\boldsymbol{\beta}} = \hat{\mathbf{q}} \tag{5}$$

where we denote the variables related to the second sequence using a similar notation with a hat ($\hat{\cdot}$). Since $\boldsymbol{\Theta}$ is an orthogonal matrix, the coefficient vectors $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}$ should be identical in theory.

If there are P moving scene points seen by the two cameras, we get a coefficient matrix $\mathbf{M} = [\boldsymbol{\beta}_1\ \ldots\ \boldsymbol{\beta}_P\ \hat{\boldsymbol{\beta}}_1\ \ldots\ \hat{\boldsymbol{\beta}}_P]$, of which the rank is bounded by P when $3K > 2P$. The rank of $\mathbf{M}$ would seem to be an appropriate metric for determining synchrony: when the F-frame fragments are temporally aligned, the coefficient correspondences are consistent with an underlying interpretation of spatiotemporal structure (the changing configurations of P points moving nonrigidly in the scene) and the rank is low. Therefore, the two sequences can be synchronized by examining the rank of $\mathbf{M}$ for various time offsets between the sequences. Let $S_1 = \{I_1(1), I_1(2), \ldots, I_1(N_1)\}$ and $S_2 = \{I_2(1), I_2(2), \ldots, I_2(N_2)\}$ be two video sequences N_1 and N_2 frames long, respectively, recorded from independently moving cameras. S_1 denotes the reference sequence and S_2 the observed sequence. The verifiable integral offset Δ is in the range of $R = [-N_1 + F, N_2 - F]$.

Till now, there are three assumptions taken into consideration when using the rank constraint.

1) The point correspondences across sequences are known.
2) The feature points are tracked throughout both sequences.
3) The temporal relationship between sequences is described by a constant offset model.

Here, we discuss only the first assumption and the other two will be discussed in the following section.

The rank constraint still holds when the correspondences across sequences are not available, if instead we use a weaker assumption [5], [6]: every 3D point tracked in the observed sequence $\hat{\mathbf{X}}^{(t)}_i$, $1 \leq i \leq P_2$ can be described as a linear combination of a subset of the 3D points tracked in the reference sequence

$$\begin{bmatrix}\hat{\mathbf{X}}^{(1)}_1 & \ldots & \hat{\mathbf{X}}^{(1)}_{P_2}\\ \vdots & \ddots & \vdots\\ \hat{\mathbf{X}}^{(F)}_1 & \ldots & \hat{\mathbf{X}}^{(F)}_{P_2}\end{bmatrix} = \begin{bmatrix}\mathbf{X}^{(1)}_1 & \ldots & \mathbf{X}^{(1)}_{P_1}\\ \vdots & \ddots & \vdots\\ \mathbf{X}^{(F)}_1 & \ldots & \mathbf{X}^{(F)}_{P_1}\end{bmatrix}[Q_1\ \ldots\ Q_{P_2}] \tag{6}$$

where the coefficients $Q_i \in \mathbb{R}^N$ of the linear combination are unknown but fixed throughout the F-frame fragment and P_1 is the number of points tracked in the reference sequence. The assumption is based on the following observation: given a set of four or more noncoplanar 3D points in a rigid body, all points in the set can be described as a linear combination of just four of the points. The affine representation of these points does not change if the same nonsingular linear transformation (e.g., translation, rotation, and scaling) is applied to all the points [27]. Combining (3) and (6) and multiplying both sides with $\boldsymbol{\Theta}^\top$, we have

$$[\hat{\boldsymbol{\beta}}_1\ \ldots\ \hat{\boldsymbol{\beta}}_{P_2}] = [\boldsymbol{\beta}_1\ \ldots\ \boldsymbol{\beta}_{P_1}][Q_1\ \ldots\ Q_{P_2}]. \tag{7}$$

Thus, the rank of the new coefficient matrix $\overline{\mathbf{M}} = [\boldsymbol{\beta}_1\ \ldots\ \boldsymbol{\beta}_{P_1}\ \hat{\boldsymbol{\beta}}_1\ \ldots\ \hat{\boldsymbol{\beta}}_{P_2}]$ is still bounded by P_1 when $3K > P_1 + P_2$. The upper bound is usually not tight, depending on the rigidity of the P_1 scene points. For example, when there are more than four points coming from the same rigid body and four of them are not coplanar, a new lower bound of the rank exists. Still, we expect the rank of the matrix $\overline{\mathbf{M}}$ to decrease in the synchronized case at least as much as it decreases in the unsynchronized case. The relaxation not only avoids point correspondence estimation between views but also enables our algorithm to handle extreme cases under wide baseline viewing condition (i.e., the cameras are situated on the opposite sides of the scene and facing each other; the cameras observe the same object, but can never see the same point).

In practice, however, due to noise, the matrix $\overline{\mathbf{M}}$ will almost be of full rank. Even without noise, considering that the

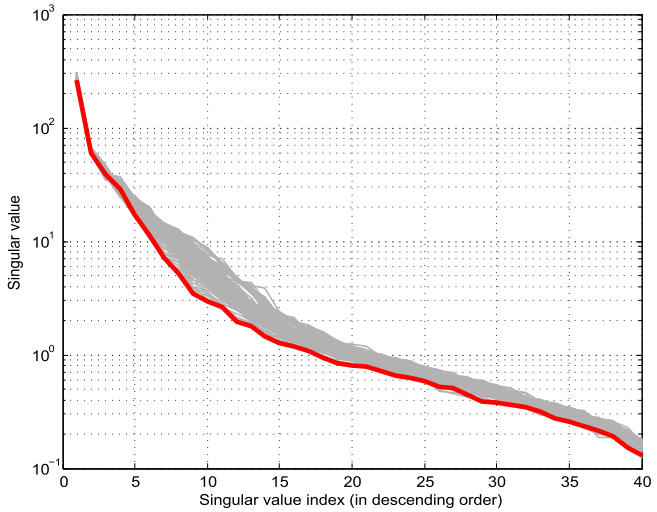


Fig. 2. Example of the singular values of $\overline{\mathbf{M}}$ in the synchronized case (red curves) and the nonsynchronized cases (gray curves). We set $P_1 = 20$ and $P_2 = 20$ in this example.

coefficient vector $\boldsymbol{\beta}$ is a least square solution of an overconstrained system, the coefficient vectors $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}$ would not be identical. To deal with these, we examine the *effective rank* $\hat{n}$ of the coefficient matrix [6]. The singular values $s_1, \ldots, s_h$ of $\overline{\mathbf{M}}$ can be obtained using singular value decomposition. We set $\hat{n} = \arg\min_j\{\sum_{k=1}^{j} s_k > \theta\}$ for a threshold θ (we use $\theta = 0.99\sum_{k=1}^{h} s_k$). Then the sum of remaining singular values, denoted by $dst = \sum_{k=\hat{n}+1}^{h} s_k$, can be used to measure the matching of two sequences. An example of the singular values of $\overline{\mathbf{M}}$ in the synchronized case (red curves) and the nonsynchronized cases (grey curves) is shown in Fig. 2. In general, the red curve has faster speed of decline compared with the grey curves. Furthermore, we transform dst to a normalized cost c by

$$c(\overline{\mathbf{M}}_\Delta) = 1 - \exp\left(-\frac{dst(\overline{\mathbf{M}}_\Delta)}{\sigma^2}\right). \tag{8}$$

Finally, we have the following optimization over integral time offset Δ:

$$\Delta^\star = \arg\min_{\Delta}\ c(\overline{\mathbf{M}}_\Delta). \tag{9}$$

IV. Nonlinear Temporal Alignment

The synchronization procedure described above is based on having image points tracked throughout both sequences and the constant time offset model assumption. To make it more robust, we present a graph-based alignment algorithm for nonlinear temporal mapping that copes with incomplete image point trajectories and outliers.

As the basis of our alignment algorithm, we use the cost c, described in the last section, to estimate the alignment quality for all possible pairs of F-frame fragments. In practice, we divide both sequences to small continuous fragments (small windows in time) of length up to F frames. We set the reference frame as the middle frame of each fragment and compute the cost c_{jk} for fragment pair $(f_1(j), f_2(k))$, where

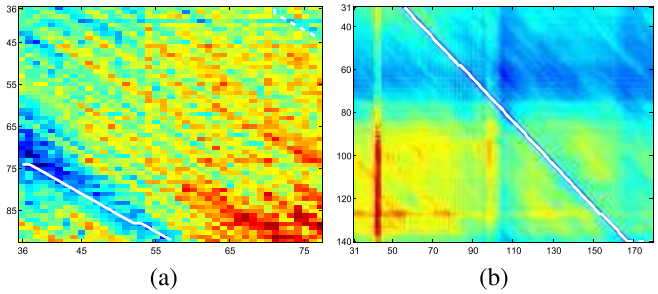


Fig. 3. Estimated temporal alignment (solid curve) and the false alignment (dashed curve). (a) Shorter path in the top-right corner of the cost matrix is a false solution. (b) Longer path covers the other path entirely. The matrix shows the alignment cost for fragment $f_1(j)$ from the reference sequence (horizontal axis) and fragment $f_2(k)$ from the second sequence (vertical axis).

$f_1(j)$ denotes the fragment with reference frame $I_1(j)$ and $f_2(k)$ the fragment with reference frame $I_2(k)$. Subsequently, we obtain a cost matrix $\mathbf{C} \in \mathbb{R}^{(N_2-2*\lfloor F/2\rfloor)\times(N_1-2*\lfloor F/2\rfloor)}$, where $\lfloor\cdot\rfloor$ denotes the floor function.

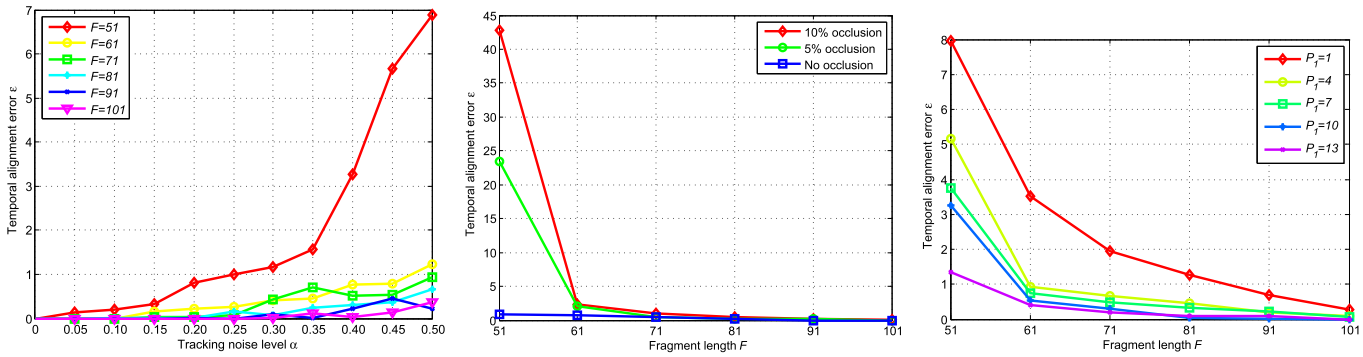
The linear relationship described in (6) is based on constant time offset assumption. When slight frame drops exist or the frame rate ratio approaches one, however, the equation can still be satisfied approximately when the reference frames of two fragments are synchronized.

In addition, the point trajectories just spanning the complete F frames can be used by our algorithm. However, care should be taken for the following two cases.

1) For each sequence, the number of points used for alignment should be identical for all fragments. Let $P_1(j)$ denote the number of points tracked throughout fragment $f_1(j)$, then $P_1 = \min\{P_1(j)\}$, $\lfloor F/2\rfloor + 1 \le j \le N_1 - \lfloor F/2\rfloor$. Similar definition for P_2.
2) Two inequations should be satisfied: $2F \ge 3K$ and $3K > P_1 + P_2$. The former is to make sure an overconstrained system for point trajectory reconstruction. The latter is to make sure that the rank of the matrix $\overline{\mathbf{M}}$ is bounded by P_1.

The discrete representation of a temporal mapping ω is referred as a *path* through the cost matrix. To support nonlinear alignment of clips with partial temporal overlap, as in [24], a set of paths from any start frame to any end frame in either video is computed. We do not use a predefined minimum threshold for path length to exclude trivial solutions; instead, we divide all the feasible paths into two pools according to the path ends in S_1 or S_2 and propose two alternatives, respectively (with the lowest normalized cost). To prevent false alignment, we finally select the longer one when a corner of $\mathbf{C}$ has a low cost [Fig. 3(a)] or their overlapping [Fig. 3(b)].

Before we apply the nonlinear temporal alignment algorithm, we first use feature tracker to generate image point trajectories. These trajectories can be short or long. Next, the 3D trajectories of the points tracked throughout each fragment are reconstructed. These two steps are performed for each sequence separately. The only step that involves combinations of sequences is computing the alignment cost according to (8) for each feasible fragment pair $f_1(j)$ and $f_2(k)$. We randomly select P_1 and P_2 points among the trajectories that span the



(a) Tracking error, $P_1 = P_2 = 13$ (b) Occlusion, $\alpha = 0.3$, $P_1 = P_2 = 13$ (c) Point number, $\alpha = 0.3$, $P_1 = P_2$

Fig. 4. Comparisons of robustness for different fragment lengths with regard to (a) tracking error, (b) occlusion, and (c) point number.

complete fragment $f_1(j)$ and $f_2(k)$, respectively. A spread-out distribution of points is desired and the point with longer trajectory gets higher priority. After the procedure is repeated T times, we choose the median as the final cost c_{jk}. The more times the procedure is repeated, the more robust our algorithm is to outliers (i.e., the points that are not tracked properly or do not fit the linear combination assumption). The computational complexity of the alignment algorithm is $O(N_1 \cdot N_2 \cdot T)$, so there is a tradeoff between accuracy and computational efficiency. We choose T according to the tracking results and the number of points that have been tracked. In general, the better the targets are tracked and the less the points have been tracked, the smaller T we choose.

V. Experiments

In this section, we evaluate the proposed framework using both synthetic trajectories and real video sequences. Given the ground truth of the synchronization mapping $\{\hat{\omega}(k), k\}_{k=1\ldots N_2}$, we use the average absolute temporal alignment error ε as our basic accuracy metric

$$\varepsilon = \frac{1}{N_2} \sum_{k=1}^{N_2} |\hat{\omega}(k) - \omega(k)|. \tag{10}$$

If the observed sequence is partially contained within the reference sequence, we consider only the overlapping.

A. Simulation

For synthetic data evaluation, we generate sequence pairs from 3D motion capture data [26] by projecting the 3D trajectories of 13 moving points onto varying image planes using synthetic camera projection matrices. Therefore, the image point trajectories in different video sequence correspond to the same set of points in 3D, and they are all throughout the whole sequences. Accordingly, the parameter T during alignment is set to 1. To enhance the reconstructibility using the original DCT basis [26], a pseudorandom number generator is used to simulate the independent camera motion. We randomly take several frames from the second sequence that has been already offset by an integer value at a maximum rate 5%. Each experiment is repeated ten times with random camera motion.

1) Robustness: For robustness evaluation, we test with tracking errors, missing data, and different numbers of points. If not specially specified, we consider the situations without missing data.

Most automatic trackers have difficulties maintaining accurate positions for all tracked points over time, especially in dynamic scenes. Fig. 4(a) shows the temporal alignment error for varying tracking noise levels and different fragment lengths. In general, the more frames used for reconstruction, the better temporal alignment our algorithm achieves. However, longer fragment results in a smaller cost matrix and accordingly a shorter path found for alignment. Missing samples occurs in practice due to occlusion, self-occlusion, or measurement failure. Fig. 4(b) shows the temporal alignment error for varying amounts of occlusion (0%, 5%, and 10% of each sequence) and different fragment lengths. As long as the visibility of a point in a sequence is sufficient to enable an overconstrained system of equations, the alignment is robust to moderate occlusion. Fig. 4(c) evaluates robustness to different numbers of points. One point may synchronize two sequences if its movement is fast and random; however, more points locating on different rigid objects will boost the chance of getting more accurate alignment. The results confirm this observation. We set K to 30 in the experiment.

2) Accuracy: We compare our algorithm with an existing technique [1], [17] that focuses on synchronizing videos containing dynamic moving objects from independently moving cameras with a large baseline. For comparing our technique with [1], we average the outputs of three different sets of five points. For comparing our technique with [17], we use the rank constraint for the perspective model. Note that their original algorithms estimate a linear time warp. For fairness, we use their computed cost matrices and run our optimal path search algorithm to estimate a nonlinear time warp.

Since the image points across sequences actually correspond to the same 3D points, we additionally compare with the following baseline: calculating the alignment quality for each frame pair using Euclidean distance between the corresponding reconstructed points in 3D. For exact corresponding frames between two video sequences, the corresponding reconstructed points should coincide. If there are no exact corresponding frames (noninteger time delay), the corresponding frames

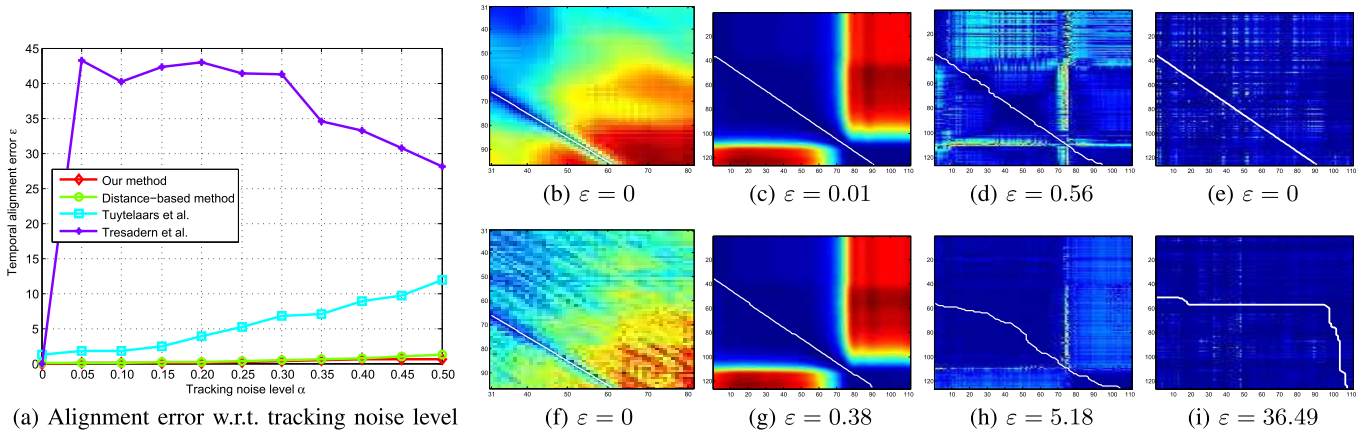


Fig. 5. (a) Quantitative comparisons of alignment accuracy using different methods regarding tracking noise levels. (b)–(e) Qualitative comparisons of cost matrix and temporal alignment (white curve) using our method, the distance-based method, and the methods by Tuytelaars and Van Gool [1] and Tresadern and Reid [17] without tracking error. (f)–(i) Idem as (b)–(e) with only tracking noise level $\alpha = 0.30$.

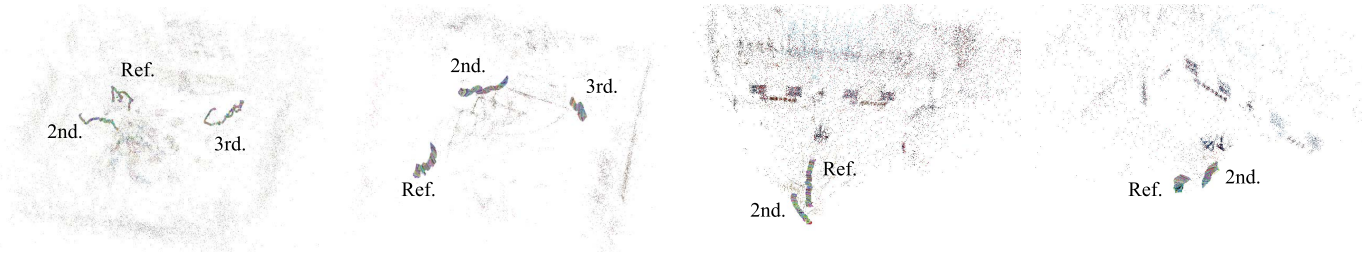


Fig. 6. 3D reconstruction results (static points in the scene and cameras). Left to right: *Block building* scene, *Exercise mat* scene, *Basketball* scene (first sequence pair), and *Basketball* scene (second sequence pair).

can be found by looking for the minimum distance. The distance-based method can be thought of as a special case of our general framework, relying on a different similarity measure. We set $F = 61$, $K = 30$, and $P_1 = P_2 = 13$ for our approach.

Fig. 5(a) compares the alignment accuracy regarding tracking errors in the stand-and-walk scene. The tracking noise in pixel is multiplying the tracking noise level α by a pseudorandom value drawn from a standard normal distribution. Fig. 5(b)–(i) illustrates qualitative comparisons for each method on a linear time warping case with $\Delta = 35$ in the faint scene. Different from ours, the dimensions of the cost matrices computed by other methods are $N_2 \times N_1$. Subsequently, the lengths of estimated paths are different from ours. Due to the sensitivity to tracking error, previous methods deteriorate at a faster rate as the tracking noise level increases compared with ours. Note that the alignment error of [17] seems not to be proportional to the tracking noise level, which we attribute to the nonlinear temporal relationship assumption. Compared with a parametric model, the nonparametric model dramatically enlarges the dimension of the solution space, which results in the poor performance when the input measurements are corrupted by noise. The tolerable performance of the distance-based method makes it an alternative when the point correspondences across sequences are available.

B. Real Data

We test our algorithm on real video sequences captured by first person cameras as shown in Fig. 1. Our data set consists of three social interaction scenes. The scenes, *Block building* and *Exercise mat*, capture tetradic interactions between children aged 5–6. For the *Basketball* scene, the players strategically take advantage of team formation (5v5). Two unsynchronized sequence pairs from each scene are used for evaluation. These sequences are 5–10 s long, with camera's translation (about 3–12 m) and rotation (about 20°–60° on the camera optical axis).

During shooting, all the cameras are set to the same recording mode. We utilize FFmpeg to extract frames from raw footage at a specified frame rate. The image sequences captured by different cameras and extracted with different frame rates are used for alignment. If not specially specified, we use 48 and 46 frames/s for the reference sequence and the observed sequence, respectively. Ground truth is provided by marking multiple frames using a photoflash before, during and after the actual recording. The remaining frames are synchronized manually.

The camera pose registration in 3D is based on structure from motion as described in [28]–[30]. The reconstruction results for different scenes are shown in Fig. 6. For the *Block building* and *Exercise mat* scenes, compared with the first sequence pair (i.e., reference and second), the cameras of the second pair (i.e., reference and third) have greater difference in viewpoint and they are basically situated on the opposite sides of the scene.

We use two-granularity tracking [31] to generate image point trajectories for each sequence. The advantage of the tracking algorithm is that it can output a mount of point

Fig. 7. Synchronization results of the *Block building* scene. Left to right: sample frames from the reference video, corresponding frames from the second video (top), and from the third video (bottom) by our method with automatically tracked point trajectories, the distance-based method, and the methods by Tuytelaars and Van Gool [1], Tresadern and Reid [17], Dexter *et al.* [18], and Wang *et al.* [24]. The points used by each method are superimposed. The blank space in the synchronization results indicates that the corresponding aligned frame is not found.

trajectories mainly locating on the trunk body, which provides good support for the linear combination assumption as described in (6). In practice, we prune the point trajectories that belong to the same moving objects and whose length is too short.

The theory of reconstructibility [26] states that it is possible to reconstruct 3D point trajectories using DCT basis precisely if a camera trajectory is fast and random. When the camera motion is slow, the camera trajectory is likely to be represented well by the DCT basis, which results in low reconstructibility and accordingly poor temporal alignment. To enhance reconstructibility, one way is to use the specialized DCT basis set, which is a projection of the original DCT onto the null space of the camera trajectories. Here, we take advantage of the remaining video sequences for each scene since a collection of asynchronous images can be interpreted as the random motion of a camera center. Correspondences of moving points across sequences are obtained manually, which are used only for trajectory reconstruction.

We compare the alignment accuracy against all the methods used in simulation. The corresponding image point trajectories required by these methods are the anatomical joints labeled manually. In addition, we compare with a 2D motion-based method [18] and a 2D feature matching-based method [24]. Note that for a better comparison, our algorithm is given three different types of input, which are manually labeled point trajectories, automatically tracked point trajectories, and both. Quantitative results of different methods are summarized in Table I. Figs. 7–10 show synchronization results for sample frames of different scenes. We set $K = 30$ and $F = 81$ for our method.

Fig. 8. Synchronization results of the *Exercise mat* scene. Idem as Fig. 7.

TABLE I

QUANTITATIVE COMPARISONS OF ALIGNMENT ERROR ON REAL SCENES. WE USE #1 AND #2 TO DENOTE THE FIRST SEQUENCE PAIR AND THE SECOND SEQUENCE PAIR OF EACH SCENE, RESPECTIVELY

	block		*mat*		*basketball*	
	#1	#2	#1	#2	#1	#2
Our method (labeled)	**0.45**	**1.27**	2.52	2.76	3.07	1.12
Our method (tracked)	0.52	1.74	**1.35**	**1.48**	**2.84**	1.54
Our method (both)	0.56	1.40	2.07	1.99	3.75	**0.92**
Distance-based method (labeled)	0.85	2.53	2.96	4.60	4.29	1.49
Tuytelaars et al. [1] (labeled)	36.91	9.16	12.05	15.63	16.81	12.42
Tresadern et al. [17] (labeled)	25.15	32.37	57.48	62.60	50.44	29.83
Dexter et al. [18] (tracked)	11.81	21.70	22.17	9.44	17.68	22.78
Wang et al. [24] (SIFT)	155.75	196.56	132.06	202.50	9.71	31.74

The proposed approach shows excellent performance in comparison with previous methods on the real scene data set. The other methods basically fail in such challenging scenarios. The content-based snapping [24] assumes that two frames are more likely to be alignable if they contain a large number of similar features and is unable to accurately synchronize sequences in the wide baseline viewing condition, except for the first sequence pair of the *Basketball* scene in which the sequences share similar content. Note that for the *Exercise mat* scene, there are certain actions performed repeatedly, which may cause ambiguity in the temporal mapping for the instant spatial-configuration-based alignment methods [1], [17]. As the effect of input on our algorithm, inadequate tracked points and gross outliers lead to the degradation in the alignment accuracy. In this case, adding accurate manual labeling increases the chance of finding a good alignment. Furthermore, if the points move slowly or smoothly, the solution tends to deviate highly from the ground truth.

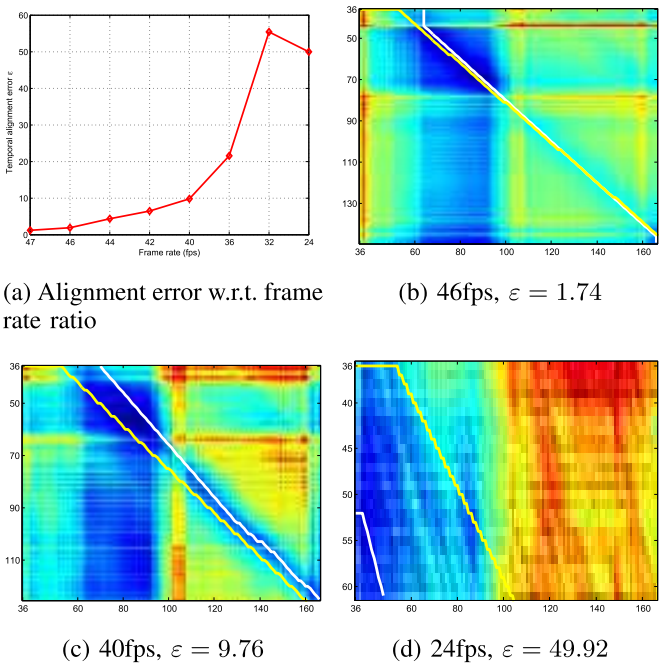
As stated previously, our alignment framework can align only video sequences with the same frame rate or different but close ones. The rank constraint becomes weakened with increasing difference in frame rate. We compare alignment

Fig. 9. Synchronization results of the *Basketball* scene (the first sequence pair). Left to right: sample frames from the reference video, corresponding frames from the second video by our method with automatically tracked point trajectories, the distance-based method, and the methods by Tuytelaars and Van Gool [1], Tresadern and Reid [17], Dexter *et al.* [18] and Wang *et al.* [24]. The points used by each method are superimposed.

Fig. 10. Synchronization results of the *Basketball* scene (the second sequence pair). Idem as Fig. 9.

accuracy with different frame rate ratios using the second sequence pair of the *Block building* scene [Fig. 11(a)]. The frame rate of the reference sequence is set to 48 frames/s, and we only change the frame rate of the observed sequence. The cost matrices computed with different frame rate ratios are shown in Fig. 11(b)–(d). When the frame rate ratio grows to 2, the rank constraint becomes insignificant.

Our current MATLAB implementation consists of parallelized but unoptimized code. Excluding the preprocessing steps (i.e., the camera pose registration in 3D and the image point trajectory generation), the entire processing stage, which consists of 3D trajectory reconstruction, computing cost matrices, and searching shortest path, takes on average 453 ms per 640×480 frame on a modern desktop computer (Intel 3.20 GHz i5-4570, four cores). Most of this running time (429 ms) is spent on 3D trajectory reconstruction, which is performed individually for each fragment pair. However, if the point correspondences across sequences are available, much faster processing time could be achieved by implementing the distance-based method, as for each moving scene point, the 3D reconstruction needs only to be performed only once per sequence. The average execution time for the entire processing stage and 3D trajectory reconstruction can be reduced to 3.4 and 2.8 ms per frame, respectively. In addition, the cost for the preprocessing steps takes on average 204 s per frame and



(a) Alignment error w.r.t. frame rate ratio

(b) 46fps, $\varepsilon = 1.74$

(c) 40fps, $\varepsilon = 9.76$

(d) 24fps, $\varepsilon = 49.92$

Fig. 11. (a) Comparisons of alignment accuracy with different frame rate ratios. The vertical axis denotes the frame rate of the second sequence. (b)–(d) Cost matrices computed when the frame rates are 46, 40, and 24, respectively, with the estimated alignment path (white) and the ground truth (yellow).

most of this running time (195 s) is spent generating image point trajectories; if needed, much faster preprocessing time could be achieved by replacing the two-granularity tracking algorithm with a different feature tracking algorithm.

VI. Conclusion

We present a general framework for synchronizing dynamic scenes in the presence of independent camera motion. We demonstrate that the coefficients from 3D trajectory reconstruction reflect the rhythm in motion and define a rank-based constraint for nonlinear temporal alignment. We fold the rank constraint into a graph-based search algorithm and compute the globally optimal path that minimizes both spatial and temporal misalignments. The main advantage of the framework is that we do not impose restrictive assumptions as complete trajectories along time or known point correspondences across sequences. Thus, we can perform the synchronization task even when the sequences are captured from distant viewpoints.

In this paper, we assume that the two image sequences correspond to the same dynamic event. The method will be exploited in future work to address other tasks, such as human action recognition or video retrieval, for which the assumption has to be relaxed.

Acknowledgment

The authors would like to thank the kindergarten affiliated to NPU and the Sports Department of NPU for their assistance with data collection.

References

[1] T. Tuytelaars and L. Van Gool, "Synchronizing video sequences," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, vol. 1, pp. 762–768.

[2] Y. Caspi and M. Irani, "Alignment of non-overlapping sequences," in *Proc. Int. Conf. Comput. Vis.*, 2001, vol. 2, pp. 76–83.

[3] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1409–1424, Nov. 2002.

[4] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," *Int. J. Comput. Vis.*, vol. 68, no. 1, pp. 53–64, 2006.

[5] L. Wolf and A. Zomet, "Correspondence-free synchronization and reconstruction in a non-rigid scene," in *Proc. Workshop Vis. Modelling Dyn. Scenes*, Jun. 2002, pp. 1–19.

[6] L. Wolf and A. Zomet, "Wide baseline matching between unsynchronized video sequences," *Int. J. Comput. Vis.*, vol. 68, no. 1, pp. 43–52, Jun. 2006.

[7] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2003, vol. 2, pp. 939–945.

[8] Y. Ukrainitz and M. Irani, "Aligning sequences and actions by maximizing space-time correlations," in *Proc. ECCV*, May 2006, pp. 538–550.

[9] C. Dai, Y. Zheng, and X. Li, "Accurate video alignment using phase correlation," *IEEE Signal Process. Lett.*, vol. 13, no. 12, pp. 737–740, Dec. 2006.

[10] A. Ravichandran and R. Vidal, "Video registration using dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 158–171, Jan. 2011.

[11] M. Singh, I. Cheng, M. Mandal, and A. Basu, "Optimization of symmetric transfer error for sub-frame video synchronization," in *Proc. ECCV*, Oct. 2008, pp. 554–567.

[12] D. Pundik and Y. Moses, "Video synchronization using temporal signals from epipolar lines," in *Proc. ECCV*, Sep. 2010, pp. 15–28.

[13] F. Pádua, R. Carceroni, G. Santos, and K. Kutulakos, "Linear sequence-to-sequence alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 304–320, Feb. 2010.

[14] C. Lu and M. Mandal, "A robust technique for motion-based video sequences temporal alignment," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 70–82, Jan. 2013.

[15] C. Lei and Y.-H. Yang, "Tri-focal tensor-based multiple video synchronization with subframe optimization," *IEEE Trans. Image Process*, vol. 15, no. 9, pp. 2473–2480, Sep. 2006.

[16] A. Yilmaz and M. Shah, "Matching actions in presence of camera motion," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 221–231, 2006.

[17] P. A. Tresadern and I. D. Reid, "Video synchronization from human motion using rank constraints," *Comput. Vis. Image Understand.*, vol. 113, no. 8, pp. 891–906, Aug. 2009.

[18] E. Dexter, P. Pérez and I. Laptev, "Multi-view synchronization of human actions and dynamic scenes," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 122:1–122:11.

[19] P. Sand and S. Teller, "Video matching," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 592–599, 2004.

[20] J. Serrat, F. Diego, F. Lumbreras, and J. M. Álvarez, "Synchronization of video sequences from free-moving cameras," in *Pattern Recognition and Image Analysis: Third Iberian Conference, IbPRIA 2007, Girona, Spain, June 6-8, 2007, Proceedings, Part II* (Lecture Notes in Computer Science), vol. 4478. Berlin, Germany: Springer, Jun. 2007, pp. 620–627.

[21] F. Diego, D. Ponsa, J. Serrat, and A. M. López, "Video alignment for change detection," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1858–1869, Jul. 2011.

[22] F. Diego, J. Serrat, and A. M. López, "Joint spatio-temporal alignment of sequences," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1377–1387, Oct. 2013.

[23] G. D. Evangelidis and C. Bauckhage, "Efficient subframe video alignment using short descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2371–2386, Oct. 2013.

[24] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung, "VideoSnapping: Interactive synchronization of multiple videos," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014, Art. no. 77.

[25] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Nonrigid structure from motion in trajectory space," in *Proc. NIPS*, 2008, vol. 1. no. 2, pp. 41–48.

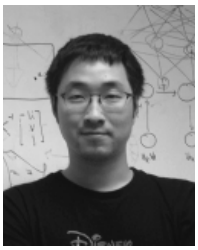
[26] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3D reconstruction of a moving point from a series of 2D projections," in *Proc. ECCV*, Sep. 2010, pp. 158–171.
[27] K. N. Kutulakos and J. Vallino, "Affine object representations for calibration-free augmented reality," in *Proc. IEEE Virtual Reality Annu. Int. Symp. (VRAIS)*, Mar./Apr. 1996, pp. 25–36.
[28] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.
[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[30] H. S. Park, E. Jain, and Y. Sheikh, "3D gaze concurrences from head-mounted cameras," in *Advances in Neural Information Proceeding Systems*. 2012, pp. 422–430.
[31] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi, "Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions," in *Proc. ECCV*, 2012, pp. 552–565.

Xue Wang received the B.E. degree in computer science and technology from Northwestern Polytechnical University (NPU), Xi'an, China, in 2007, and the M.E. degree in computer applied technology from Capital Normal University, Beijing, China, in 2010. She is currently pursuing the Ph.D. degree with the School of Computer Science, NPU.

Her research interests include object tracking and human behavior analysis.

Jianbo Shi received the B.A. degree from Cornell University, Ithaca, NY, USA, in 1994, and the Ph.D. degree in computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1998.

He joined the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, in 1999, as a Research Faculty Member. He then joined the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA, in 2003, where he is currently a Professor. His long-term interests center around a broader area of machine intelligence—a visual thinking module that allows computers not only to understand the environment around us but also to achieve higher level cognitive abilities, such as machine memory and learning. His current research interests include human behavior analysis and image recognition-segmentation.



Hyun Soo Park received the B.S. degree from the Pohang University of Science and Technology, Pohang, South Korea, in 2007, and the M.S. degree and the Ph.D. degree in mechanical engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2009 and 2014, respectively.

He is currently a Post-Doctoral Fellow with the University of Pennsylvania, Philadelphia, PA, USA. His research aims to develop computational social intelligence—an ability to perceive, model, and predict such visual social signals. His current research interests include human interaction by sending visible social signals, such as gaze movements, facial expressions, and body gestures.

Qing Wang (M'05) received the B.S. degree in mathematics from Peking University, Beijing, China, in 1991, and the M.E. and Ph.D. degrees from Northwestern Polytechnical University (NPU), Xi'an, China, in 1997 and 2000, respectively.

He joined the School of Computer Science, NPU, in 1991, where he is currently a Professor. His research interests include computer vision and computational photography.

Prof. Wang has been a member of the Association for Computing Machinery since 2009. He has been a Senior Member of the China Computer Federation (CCF) since 2005.