Spatio-Temporally Consistent Correspondence for Dense Dynamic Scene Modeling

Dinghuang Ji(✉), Enrique Dunn, and Jan-Michael Frahm

The University of North Carolina at Chapel Hill, Chapel Hill, USA
{jdh,dunn,jmf}@cs.unc.edu

Abstract. We address the problem of robust two-view correspondence estimation within the context of dynamic scene modeling. To this end, we investigate the use of local spatio-temporal assumptions to both identify and refine dense low-level data associations in the absence of prior dynamic content models. By developing a strictly data-driven approach to correspondence search, based on bottom-up local 3D motion cues of local rigidity and non-local coherence, we are able to robustly address the higher-order problems of video synchronization and dynamic surface modeling. Our findings suggest an important relationship between these two tasks, in that maximizing spatial coherence of surface points serves as a direct metric for the temporal alignment of local image sequences. The obtained results for these two problems on multiple publicly available dynamic reconstruction datasets illustrate both the effectiveness and generality of our proposed approach.

Keywords: Two-View correspondences · Motion consistency

1 Introduction

Dynamic 3D scene modeling addresses the estimation of time-varying geometry from input imagery. Existing motion capture techniques have typically addressed well-controlled capture scenarios, where aspects such as camera positioning, sensor synchronization, and favorable scene content (*i.e.*, fiducial markers or "green screen" backgrounds) are either carefully designed *a priori* or controlled online. Given the abundance of available crowd-sourced video content, there is growing interest in estimating dynamic 3D representations from uncontrolled video capture. Whereas multi-camera static scene reconstruction methods leverage photoconsistency across spatially varying observations, their dynamic counterparts must address photoconsistency in the spatio-temporal domain. In this respect, the main challenges are (1) finding a common temporal reference frame across independent video captures, and (2) meaningfully propagating temporally

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46466-4_1) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VI, LNCS 9910, pp. 3–18, 2016.
DOI: 10.1007/978-3-319-46466-4_1

varying photo-consistency estimates across videos. These two correspondence problems – temporal correspondence search among unaligned video sequences and spatial correspondence for geometry estimation – must be solved jointly when performing dynamic 3D reconstruction on uncontrolled inputs.

In this work, we address both of these challenges by enforcing the geometric consistency of optical flow measurements across spatially registered video segments. Moreover, our approach builds on the thesis that *maximally consistent geometry is obtained with minimal temporal alignment error*, and *vice versa*. Towards this end, we posit that it is possible to recover the spatio-temporal overlap of two image sequences by maximizing the set of consistent spatio-temporal correspondences (that is, by maximizing the completeness of the estimated dynamic 3D geometry) among the two video segments.

In practice, our approach addresses the spatio-temporal two-view stereo problem. Taking as input two unsynchronized video streams of the same dynamic scene, our method outputs a dense point cloud corresponding to the evolving shape of the commonly observed dynamic foreground. In addition to outputting the observed 3D structure, we estimate the temporal offset of a pair of input video streams with a constant and known ratio between their frame rates. An overview of our framework is shown in Fig. 1. Our framework operates within local temporal windows in a strictly data-driven manner to leverage the low-level concepts of local rigidity and non-local geometric coherence for robust model-free structure estimation. We further illustrate how our local spatio-temporal assumptions can be built to successfully address problems of much larger scope, such as content-based video synchronization and object-level dense dynamic modeling.

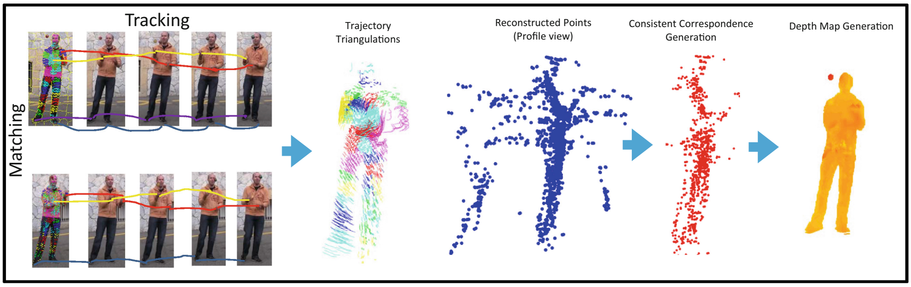

Tracking
Matching
Trajectory Triangulations
Reconstructed Points (Profile view)
Consistent Correspondence Generation
Depth Map Generation

Fig. 1. Overview of the proposed approach for dense dynamic scene reconstruction from two input video streams.

2 Related Work

For static environments, very robust structure-from motion (SfM) systems [1–3] and multi-view stereo (MVS) approaches[4] have shown much success in recovering scene geometry with high accuracy on a large variety of datasets. Modeling non-static objects with this framework, however, is considerably more difficult because the assumptions driving correspondence detection and 3D point triangulation in rigid scenarios cannot be directly applied to moving objects. To address these challenges, a wide array of dynamic scene reconstruction techniques have been introduced in the computer vision literature, in capture situations that are controlled or uncontrolled, synchronized or unsynchronized, single-view or multi-view, and model-based or model-free.

In general, highly controlled image capture scenarios have shown considerable success for non-static scene capture because they are able to leverage more powerful assumptions w.r.t. appearance and correspondence of scene elements. For example, Joo *et al.* [5,6] used a large-scale rig of 480 synchronized cameras arranged along a sphere to obtain high-quality, dense reconstructions of moving objects within the capture environment. For more general applications, Kim *et al.* [7] designed a synchronized, portable, multi-camera system specifically tailored for dynamic object capture. Jiang *et al.* [8] and Taneja *et al.* [9] further proposed probabilistic frameworks to model outdoor scenes with synchronized handheld cameras. Mustafa *et al.* [10] introduced a general approach to dynamic scene reconstruction from multiple synchronized moving cameras without prior knowledge or limiting constraints on the scene structure. These works, and others [11–17], clearly indicate the strong potential for non-rigid reconstruction in general capture scenarios, and they highlight in particular the usefulness of multiple synchronized video streams toward this end. In this paper, we build on these works by automatically recovering the temporal alignment of unsynchronized video streams as part of the dense, dynamic reconstruction process.

Single-view video capture can be considered as a dynamic reconstruction scenario inherently lacking the benefits of multi-view synchronization. On this front, the monocular method of Russell *et al.* [18] is most germane to our approach. The authors employ automatic segmentation of rigid object subparts, for example 3D points on the arms, legs, and torso of a human, and solve the dynamic reconstruction problem by jointly computing hierarchical object segmentation and sparse 3D motion. Their notion of *spatial consistency* of rigid subparts is an important contribution that we leverage in our approach to unsynchronized multi-view reconstruction. A key distinction is that our method utilizes multiple camera views to recover relative object translation in the static surrounding environment, which is not completely recoverable using monocular input alone.

Despite the large amount of crowd-sourced video data available on the Internet (for example, multiple video uploads from a live concert), relatively little research has focused on dynamic 3D reconstruction from unsynchronized, concurrent capture. To our knowledge, Zheng *et al.* [19] were the first to propose a solution to this interesting reconstruction task. The authors introduced a dictionary learning method to simultaneously solve the problem of video synchronization and sparse 3D reconstruction. In this method, the frame offsets of multiple videos are obtained by sparse representation of the triangulated 3D shapes, and the shapes are iteratively refined with updated sequencing information. However, this approach is not automatic, relying heavily on manually labeled correspondences on the rigid bodies, and the resulting reconstructions are relatively sparse (*i.e.*, they represent a human using only 15 3D points). Their extended version [20], further asserts that both outlier correspondences and reduced/small temporal overlap will hinder the accuracy of the temporal alignment. In contrast to Zheng *et al.* [19], our work aims to jointly recover dense object geometry and temporal information in an unsupervised manner.

In the past, multi-view geometric reasoning has been employed for the general problem of video synchronization. These methods are related to the video synchronization aspect our work, but they do not provide dense 3D geometry. For example, Basha *et al.* [21,22] proposed methods for computing partial orderings for a subset of images by analyzing the movement of dynamic objects in the images. There, dynamic objects are assumed to move closely along a straight line within a short time period, and video frames are ordered to form a consistent motion model. Tuytelaars *et al.* [23] proposed a method for automatically synchronizing two video sequences of the same event. They do not enforce any constraints on the scene or cameras, but rather rely on validating the rigidity of at least five non-rigidly moving points among the video sequences, matched and tracked throughout the two sequences. In [24], Wolf and Zomet propose a strategy that builds on the idea that every 3D point tracked in one sequence results from a linear combination of the 3D points tracked in the other sequence. This approach works with articulated objects, but requires that the cameras are static or moving jointly. Finally, Pundik *et al.* [25] introduced a novel formulation of low-level temporal signals computed from epipolar lines. The spatial matching of two such temporal signals is given by the fundamental matrix relating each pair of images, without requiring pixel-wise correspondences.

3 Spatio-Temporal Correspondence Assessment

Our goal is to analyze two spatially-registered video sub-sequences of equal length, in order to determine the largest set of spatio-temporally consistent pixel correspondences belonging to a commonly observed dynamic foreground object. In particular, we are interested in building two-view correspondence-based visual 3D tracks spanning the entire length of the sub-sequences and assessing the validity of the initial correspondences in terms of the geometric properties of the 3D tracks. Our goal has two complimentary interpretations: (1) to develop a spatio-temporal correspondence filtering mechanism, and (2) to provide a measure of local spatio-temporal consistency among video sub-sequences in terms of the size of the valid correspondence set. We explore both these interpretations within the context of video synchronization and dense dynamic surface modeling.

3.1 Notation

Let $\{\mathcal{I}_i\}$ and $\{\mathcal{I}'_j\}$ denote a pair of input image sequences, where $1 \leq i \leq M$ and $1 \leq j \leq N$ are the single image indices. For each image $\mathcal{I}_k \in \{\mathcal{I}_i\} \cup \{\mathcal{I}'_j\}$, we first obtain via structure-from-motion (SfM) a corresponding camera projection matrix, $\mathbf{P}(\mathcal{I}_k) = \mathbf{K}_k\, [\mathbf{R}_k | - \mathbf{R}_k \mathbf{C}_k]$, where $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{C}$ respectively denote the camera's intrinsic parameter matrix, external rotation matrix, and 3D position. Let $\mathbf{F}_{ij}$ denote the fundamental matrix relating the camera poses for images $\mathcal{I}_i$ and $\mathcal{I}'_j$. Furthermore, let $\mathcal{O}_i$ and $\mathcal{O}'_j$ denote optical flow fields for corresponding 2D points in consecutive images (*e.g.*, $\mathcal{I}_i \rightarrow \mathcal{I}_{i+1}$ and $\mathcal{I}'_j \rightarrow \mathcal{I}'_{j+1}$) in each of the two input sequences. Finally, let $\mathbf{x}_{ip}$ and $\mathbf{X}_{ip}$ denote the 2D pixel position and the 3D world point, respectively, for pixel p in image $\mathcal{I}_i$ (and similarly $\mathbf{x}'_{jp}$ and $\mathbf{X}'_{jp}$ for image $\mathcal{I}'_j$).

3.2 Pre-processing and Correspondence Formulation

Spatial Camera Registration. Our approach takes as input two image streams capturing the movements of a dynamic foreground actor, under the assumption of sufficient visual overlap that enables camera registration to a common spatial reference defined by a static background structure. Inter-sequence camera registration is carried out in a pre-processing step using standard SfM methods [15] over the aggregated set of frames, producing a spatial registration of the individual images from each stream. Since the goal of this stage is simply image registration of the two sequences, the set of input images for SfM can be augmented with additional video streams or crowd-sourced imagery for higher-quality pose estimates; however, this is not necessarily required for our method to succeed.

Dynamic Foreground Segmentation. SfM simultaneously recovers the camera poses for the input images and reconstructs the 3D structure of the static background. The first step in our method is to build a reliable dynamic foreground mask for each image using the available 3D SfM output. At first blush, it seems that this task can be accomplished by simply reprojecting the SfM 3D points into each image and aggregating these projections into a background mask. However, this approach is less effective for automatic foreground segmentation primarily because it does not account for spurious 3D point triangulations of the dynamic foreground object. Hence, to identify the non-static foreground points in an image, we adopt a three-stage process: First, we perform RANSAC-based dominant 3D plane fitting on the SfM point cloud, under the assumption that large planar structures will be part of the background. We iteratively detect dominant planes until we have either included over 70 % of available points or the estimated inlier rate of the current iteration falls below a pre-defined threshold. Second, for the remaining reconstructed 3D points not belonging to a dominant plane, we identify their set of nearest 3D neighbors and measure the photoconsistency of this set with their corresponding color projections into the image under consideration. We measure the normalized cross correlation (NCC) of these samples and threshold values above 0.8 as background and below 0.5 as foreground. Third, we perform a graph-cut optimization to determine a global foreground-background segmentation, where we use the points on the dominant planes along with photoconsistent reprojections as initial background seeds, while the nonphotoconsistent pixels are considered foreground seeds. Figure 2 illustrates an example of our segmentation output.

Correspondence Search Space. Consider two temporally corresponding image frames $\mathcal{I}_i$ and $\mathcal{I}'_j$. For a given pixel position $\mathbf{x}_{ip}$ contained within the dynamic foreground mask of image $\mathcal{I}_i$, we can readily compute a correspondence $\mathbf{x}'_{jp}$ in image $\mathcal{I}'_j$ by searching for the most photoconsistent candidate along the epipolar line $\mathbf{F}_{ij}\mathbf{x}_{ip}$. We can further reduce the candidate set $\Omega(\mathbf{x}_{ip}, \mathbf{F}_{ij}) \in \mathcal{I}'_j$ by only considering points along the epipolar line contained within the foreground mask of $\mathcal{I}'_j$ (Fig. 3(a) and (b)). In this manner, we have $\Omega(\mathbf{x}_{ip}, \mathbf{F}_{ij}) = \{\mathbf{x}'_{jq} \mid \mathbf{x}_{ip}\mathbf{F}_{ij}\mathbf{x}'_{jq} = 0\}$. Henceforth, we shall omit the dependence on the pre-computed camera geometry and segmentation estimates from our notation, denoting the set of candidate matches for a given pixel as $\Omega(\mathbf{x}_{ip})$. We measure NCC w.r.t. the reference pixel $\mathbf{x}_{ip}$ using 15×15 patches along the epipolar line, and all patches with a NCC value greater than 0.8 are deemed potential correspondences. Once $\Omega(\mathbf{x}_{ip})$ is determined, its elements $\mathbf{x}'_{jq}$ are sorted in descending order of their photoconsistency value. Figure 3(c) and (d) provides an example of our epipolar correspondence search for an image pair.

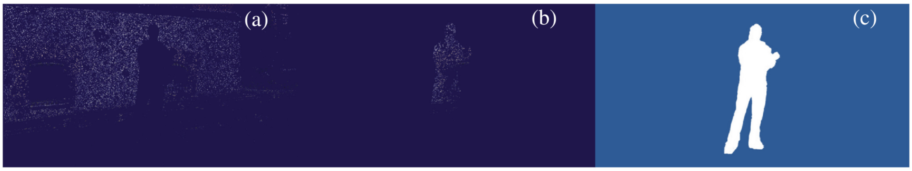


Fig. 2. (a) Background mask that has high color consistency. (b) Foreground mask with low color consistency. (c) Segmented result.

3.3 Assesement and Correction Mechanisim

Based on the example shown in Fig. 3, we propose a method to discern wrong correspondences and correct them with an alternative pixel matches. The steps of our method are as follows:

Step ❶: Building Motion Tracks. The set of 2D feature points $\{\mathbf{x}_{ip}\}$ and currently selected corresponding points $\{\mathbf{x}'_{jq}\}$ are updated with optical flow motion vectors computed between neighboring frames using the approach of Brox et al. [26]. Thus we have $\{\mathbf{x}_{i+1,p}\} = \{\mathbf{x}_{i,p}\} + \mathcal{O}_i$ and $\{\mathbf{x}'_{j+1,q}\} = \{\mathbf{x}_{jq}\} + \mathcal{O}'_j$. We select the video with the higher frame rate as the target sequence, which will be temporally sampled according to the frame rate ratio α among the sequences. The reference sequence will be used at its native frame rate. Hence, given a temporal window of W frames, the reference video frames and their features will be denoted, respectively, by $\mathcal{I}_i$ and $\{\mathbf{x}_{i,p}\}$, where $1 \leq i \leq W$, denotes the frame index. Accordingly, the frames and features in the target video frames will be denoted by $\mathcal{I}'_j$ and $\{\mathbf{x}'_{j+w*\alpha,q}\}$, where j corresponds to the temporal frame offset

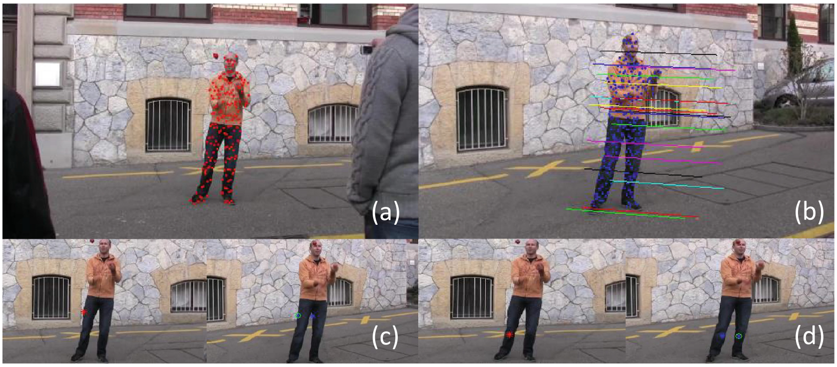


Fig. 3. (a) Local features in reference image. (b) Corresponding points are found along the epipolar lines in the target image. In (c) and (d) Red stars: Feature point in reference frame. Blue stars: Matched feature points in the target frame. Green circles: Points with highest NCC values. (c), the point with the highest NCC value is actually the correct correspondence. (d), the green circle is indicating the wrong match. The other candidate is the correct correspondence and should be used for triangulation.

between the two sequences, and $0 \leq w < W$. The size of the temporal window must strike a balance between building informative 3D tracks for spatial analysis and maintaining the reliability of the chain of estimated dense optical flows.

The initial set of correspondence estimates $\{\mathbf{x}_{ip}\}$, $\{\mathbf{x}'_{jq}\}$ are temporally tracked through successive intra-sequence optical flow estimates, and their updated locations are then used for two-view 3D triangulation. Namely, for each point $\mathbf{x_{ip}}$ selected at frame p, we have a 3D track $\mathbf{T}_i = \{\mathbf{X}_{iw}\}$ comprised of $1 \leq w \leq W$ 3D positions determined across the temporal sample window.

Step ❷: Enforcing Local Rigidity. Local rigidity assumes a pair of nearby 3D points in the scene will maintain a constant Euclidean distance throughout our temporal observation window. Assuming a correct spatio-temporal inter-sequence registration and accurate intra-sequence optical flow estimates, deviations from this assumption are attributed to errors in the initial correspondence estimation. More specifically, tracks having incorrect initial correspondences will present inconsistent motion patterns. Accordingly, the key component of our rigidity estimation is the scope of our locality definition. To this end, we use the appearance-based super-pixel segmentation method proposed in [27] to define relatively compact local regions aligned with the observed edge structure. The SLIC scale parameter is adaptively set such that the total of superpixels contained within the initial segmentation mask is 30. The output of this over-segmentation of the initial frame in the reference sequence is a clustering of our 3D tracks into disjoints partitions $\{\mathcal{C}_c\}$, where $1 \leq c \leq 30$.

Having defined disjoint sets of 3D tracks, we independently evaluate the rigidity of each track cluster. We measure this property in terms of the largest consensus set of constant pairwise distances across successive frames. Although this set can be identified through exhaustive evaluation of all pairwise track distances, we instead take a sampling approach for efficiency. We iteratively select one of the tracks in $\mathcal{C}_c$ and compare the temporal consistency against all other tracks. We then store the track with the largest support within $\mathcal{C}_c$. An outline of our sampling method is presented in Algorithm 1. Our local rigidity criteria decides if two trajectories are consistent based on the accumulated temporal variation of point-wise distance of two 3D tracks over time:

$$\sum_{i=2}^{W} \Big| \|\mathbf{X}_{m,i-1} - \mathbf{X}_{n,i-1}\|_2 - \|\mathbf{X}_{m,i} - \mathbf{X}_{n,i}\|_2 \Big|, \ \mathbf{T}_n, \mathbf{T}_m \in \mathcal{C}_c \tag{1}$$

Once the consensus track set has been identified, all its members are considered inliers to the rigidity assumption, while all tracks not belonging to the consensus set are labeled as outliers.

Algorithm 1. SAMPLING FOR LOCAL RIGIDITY TRACK CONCENSUS

```
Input: 3D trajectories $\mathbf{T}_i(m)$, $1 \leq m \leq |\mathcal{C}_i(c)|$
Output: Inliers trajectories set $\{\hat{\mathcal{C}}_i(c)\}$
iterations = 0
$\hat{\mathcal{C}}_i(c) = NULL$
while $iterations \leq |\mathcal{C}_i(c)|/5$ do
    $\mathcal{C}'_i(c) = NULL$
    Draw a random trajectories $\mathbf{T}_i(m)$
    for $k \in [1, \|\mathcal{C}_i(c)\|]$ do
        decide if $\mathbf{T}_i(m)$ and $\mathbf{T}_i(k)$ are consistent
        if consistent then
            add $k$ into $\mathcal{C}'_i(c)$; if $\mathcal{C}'_i(c) = \mathcal{C}_i(c)$ then
                return
    if $\mathcal{C}'_i(c) \geq \hat{\mathcal{C}}_i(c)$ then
        $\hat{\mathcal{C}}_i(c) = \mathcal{C}'_i(c)$
```

Step ❸: Enforcing Structural Coherence. Local rigidity in isolation is unable to determine systematic errors caused by motion correlation among content having similar appearance. A particular challenge is the presence of poorly textured and (nearly) static scene elements, as both appearance and motion cues are ill-defined in this scenario. For example, in Fig. 5(a), some correspondences are located on the left leg, while the true correspondences should be on the right leg. In order to make our correspondence estimation more robust, we further enforce the assumption of geometric coherence within local structure estimates deemed to be locally rigid. We consider two types of non-local coherence violations:

1. **Track-Bundle Consistency.** 3D Tracks emanating from a common compact image region should also correspond to a compact set of 3D trajectories. We

observe that small subsets of inlier (*i.e.*, mutually rigid) 3D tracks can be spatially disjoint from the remaining tracks belonging to the same initial cluster (Fig. 5(b)). We measure this behavior by analyzing the results of individual pairwise 3D point sampling used in step ❷for rigidity consensus estimation. We aggregate all the sampled $N = \|\mathcal{C}_c\|$ pairwise rigid distances of the inlier set into a single vector $S_c \in \mathbf{R}^N$ and sort the elements by increasing distance. We then scan for an inflection point depicting the largest pairwise deviation among successive bins in S_c and threshold on both the relative magnitude and the percentile of the inflection point location within the histogram. Inflection points found in the top and bottom 10 % quantiles are to be discarded. If an inflection point is found in the histogram, the corresponding distance value is used as a distance consistency threshold. Tracks exhibiting an average distance to other tracks greater than the consistency threshold are removed from the inlier set $\mathcal{C}_c$. Figure 4 illustrates the behavior of the distance histogram for different 3D track bundle scenarios. The above framework operates under the assumption that locally inconsistent tracks represent a small fraction of a cluster's track bundle.

2. **Inter-Cluster Consistency.** The scenario where the majority (or all) of the mutually rigid tracks within a cluster are structured outliers is extremely uncommon but cannot be identified through track-bundle consistency (Fig. 5(c)). To address this challenge, we impose thresholds on the spatial divergence between the average 3D positions of a given track and a fixed global 3D reference representative of the estimated structure across the entire image. We define this reference to be the 3D centroid of the 3D tracks of all other clusters. This approach is aimed at identifying gross outliers within the context of a single foreground dynamic object and is to be considered a special-purpose noise filtering technique. In practice, 3D tracks away from the moving body are identified and singled out as correspondence outliers.

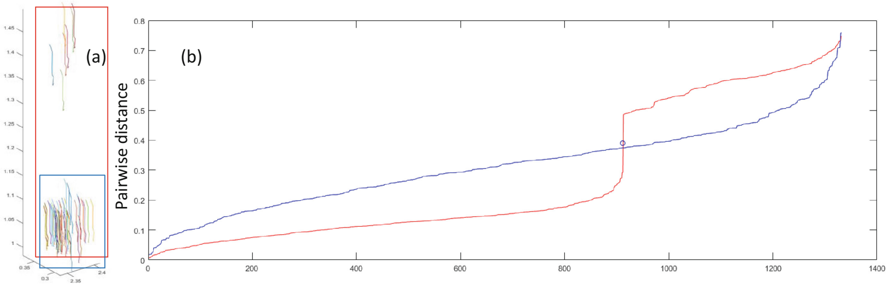


Fig. 4. In (a), trajectories from wrong correspondences deviate away from the inlier trajectories (outlined in blue). (b) The sorted pairwise distance array of all inliers has no abrupt gradient in the middle, sorted pairwise distance array of all trajectories will have those cutting edge when outlier trajectories are present.

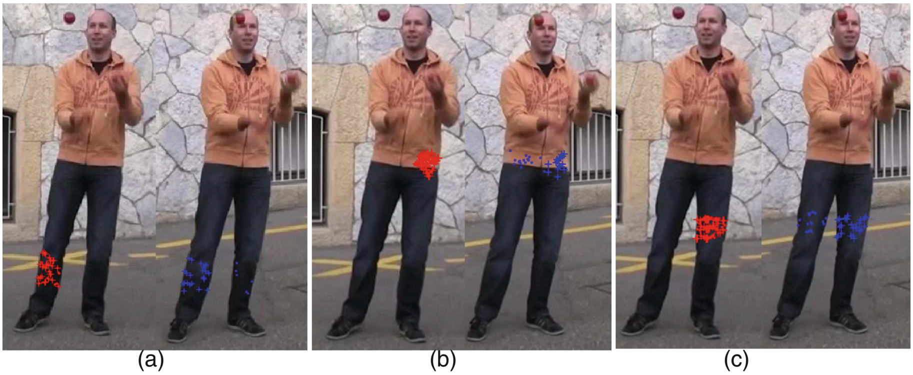


Fig. 5. Corresponding points in image pairs. Red dots (crosses): Feature (inlier) points within one super-pixel in the reference frame. Blue dots (crosses): Correspondence (inlier) points found in the target frame. In (a), outliers on the left leg are detected because they located in different rigid parts. In (b), outliers on the right waist are removed because they are far away from majority of other trajectories. In (c), correct correspondences are the minority (there might be repetitive correspondences in the target frame). The wrong correspondences are removed by the depth constraints.

Step ❹: Track Correction. The set of 3D tracks determined to be outliers by our preceding validation steps are assumed to occur due to an outlier feature correspondence $\mathbf{x}_{ip} \leftrightarrow \mathbf{x}_{jq}$. Accordingly, to correct this erroneous initial assignment, we revisit the sorted set of correspondence candidates $\Omega(\mathbf{x}_{ip})$ lying on the epipolar line. We will replace the initial assignment with the next-most photo-consistent element of $\Omega(\mathbf{x}_{ip})$ and evaluate the local rigidity of the updated 3D track across the temporal sampling window. We can now modify the correspondence to regenerate the 3D track (i.e. step ❶) and re-run our original rigidity sampling procedure (i.e. step ❷) over the entire cluster to account for possible changes to the consensus set. In practice, it is more efficient to verify the rigidity of each updated track against a small sample of the current consensus/inlier (i.e. locally rigid) set of tracks. The process is repeated until each original feature has either (1) been determined to be an inlier or (2) exhausted the candidate set.

3.4 Applications to Stream Sequencing and 3D Reconstruction

We have described a framework to determine and enhance the spatio-temporal consistency of two-view pixel correspondences across a time window. Our image-wide active correspondence correction framework effectively maximizes the number of locally consistent 3D tracks. The relevance of this functionality lies in the insight that, given an unknown temporal offset between two spatial overlapping video sequences, scanning a short video segment from one sequence over the entirety of the other sequence can be used to identify the temporal offset between those sequences. Figure 6(b) shows the average correspondences with different offsets (computed over 50 consecutive frames from one of our datasets), we can see our method obtain the highest value on the 0 offset point, which means accurate alignment. The criteria to determine alignment is, intuitively, the offset resulting maximal locally rigid (e.g. inlier) 3D tracks. Conversely, determining a robust and dense set of inter-sequence correspondences, directly provides the observed 3D geometry given knowledge of the imaging geometry. A straightforward way to generate depthmaps under our framework is to perform bi-linear 2D interpolation on each sequence frame for all inlier 3D tracks. Figure 6(a), illustrates the depthmap generated by our approach without any post-processing corrections.

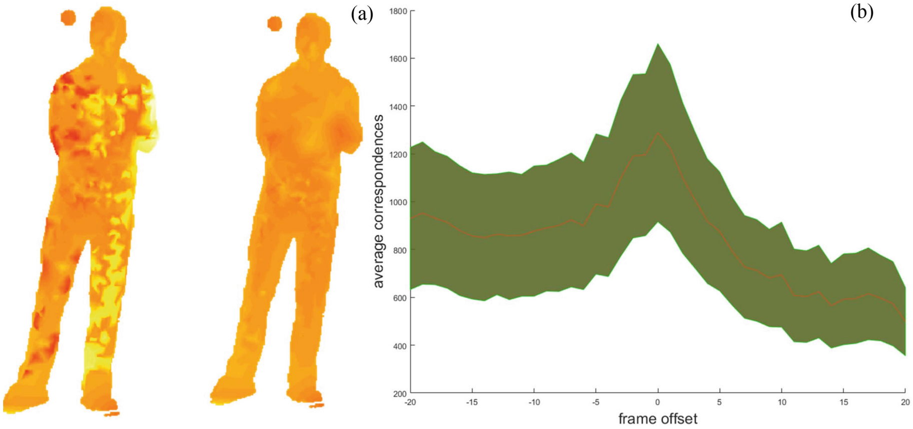


Fig. 6. (a) show depth map generated from raw correspondences (Left) and the corrected correspondences (Right). (b)Average correspondences with different offsets(red curve), the green boundary should the plus minus standard deviation.

4 Experiments

Experimental Setup. All reported experiments considered a temporal window size of $W = 6$, and unless stated otherwise, the initial correspondence set is comprised of all putative pixel correspondences along the epipolar line with an NCC value above 0.8. We evaluated our method on three datasets: the ETH juggler [28], the CMU bat [5], and the UNC juggler [19]. For the ETH dataset (6 cameras) and the UNC dataset (4 cameras), we select the pair of cameras having the smallest baseline. For the CMU dataset, we select two neighboring cameras facing the front of the players. The CMU dataset provides reconstructed 3D points which are used as ground truth to evaluate the accuracy of our estimated 3D triangulations and depth maps. The UNC dataset is not synchronized; hence, we adopt the synchronized result from [19] as sequencing ground truth. Details for each of the three considered datasets are provided in Table 1.

Table 1. Composition of our datasets.

Name	# Video frames	GT 3D Points	Synchronized	Moving Cameras	Outdoor Scene
ETH	200	No	Yes	Yes	Yes
CMU	160	Yes	Yes	No	No
UNC	150	No	No	Yes	Yes

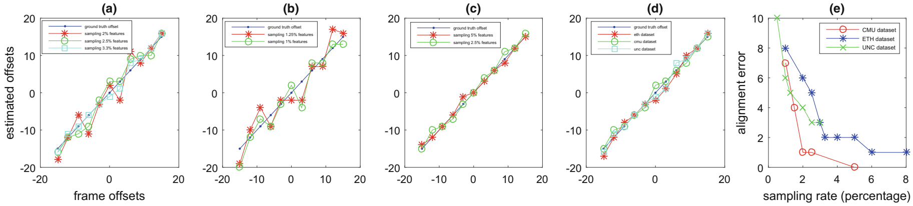
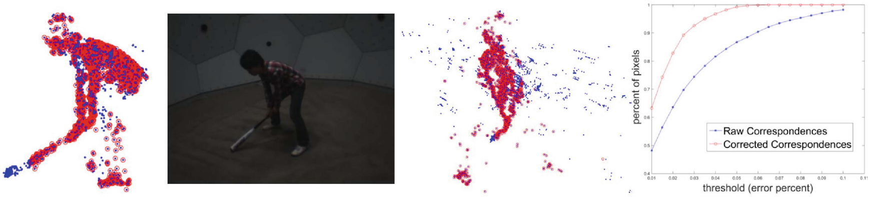


Fig. 7. Accuracy of our synchronization estimation across different datasets scenarios.

Synchronization Evaluation. In order to evaluate synchronization accuracy, we carried out experiments with temporal offsets between the reference and the target sequence in the range of $[-15, 15]$ with step size 3. We considered the following scenarios: (1) common frame with varying pixel sampling density, and (2) one sequence having double the frame rate of the other. Figure 7(a-c) shows respectively the results for ETH, UNC, and CMU datasets under varying pixel densities. By controlling the density of considered pixels within each local neighborhood (i.e. SLIC-based superpixel segmentation) we can directly control the computational burden of our sampling rigidity framework. Alternatively, we may perform KLT-based feature selection. For efficiency reasons, we simply select in these experiments a fixed number of random pixels as features for correspondence analysis within a local neighborhood $\mathcal{C}_c$. We experimented with pixel densities of 2 %, 2.5 %, and 3.3 %. The results illustrated in Fig. 7(a-c) highlight the positive effect of increased pixel densities towards synchronization



(a) (b) (c) (d)

Fig. 8. Results of corrected point cloud on the CMU dataset. Left: Blue 3D points depict the originally reconstructed 3D points from initial correspondences, while red points denote the 3D points obtained through corrected correspondences. Left middle: Corresponding reference image. Right center: A side view of the same structure. Right: Accuracy for both original and corrected point sets.

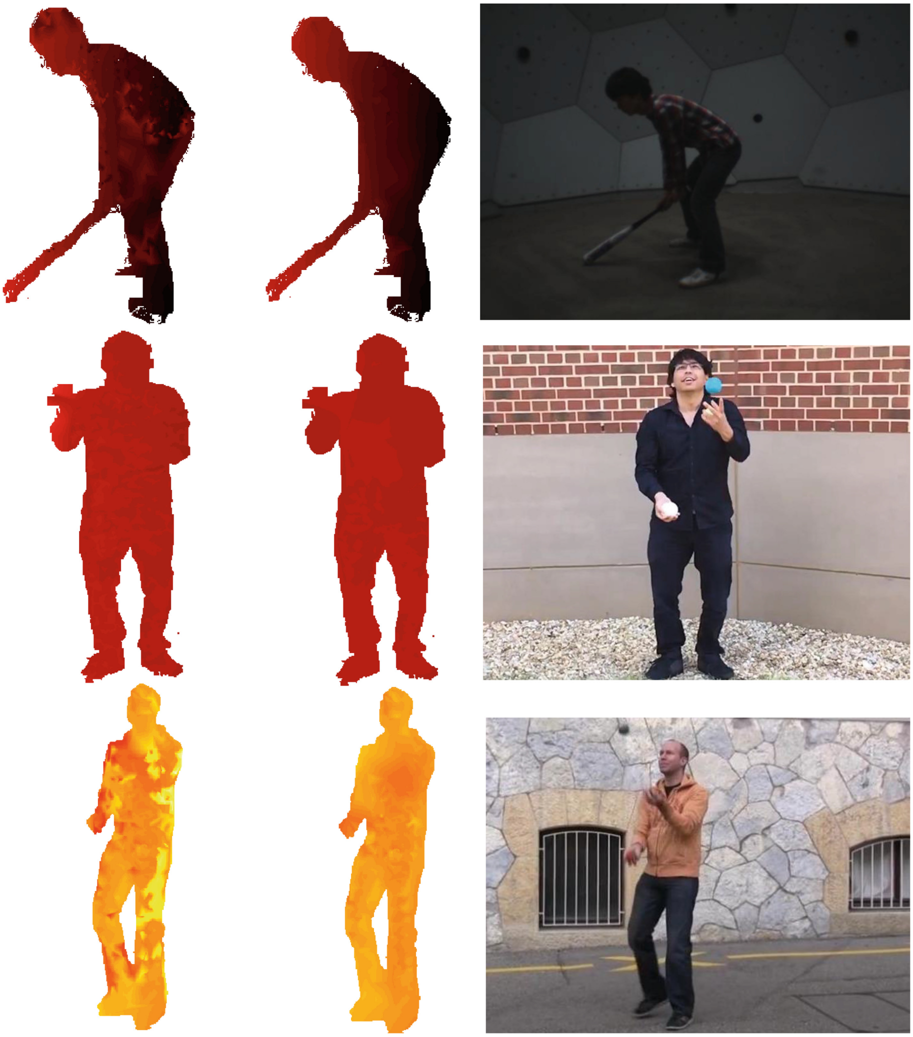


Fig. 9. Qualitative results illustrating the effectiveness of our correspondence correction functionality.

accuracy. We observe that, in addition to segments exhibiting reduced motion or poorly textured content, repetitive motion was a source of synchronization ambiguity leading to potential errors. Figure 7(d) shows the alignment results with the target sequence at twice the frame rate of reference sequence. We use 3.3 %, 1.25 %, and 5 % sampling density, and the results are very close to the equal-frame-rate test, with a decrease in average accuracy of 9 %. In Fig. 7(e) we show more synchronization results with variable sampling rates for video streams.

Dense Modeling Evaluation. We explored the effectiveness of our correspondence correction functionality when applied for 3D reconstruction. Given that the CMU dataset provides groundtruth 3D structure values, we include the reconstruction error of our 3D reconstructions. In Fig. 8(a) and (c), we show the front and back view of the estimated 3D points. We observe our method's ability to effectively remove outlier 3D structure. In Fig. 8(d), we quantitatively evaluate the accuracy of our depth map, in terms of the percentage of pixels falling within variable accuracy thresholds. Figure 9 shows some qualitative comparisons of our interpolated depth maps obtained from correspondence-corrected 3D points against the depthmaps interpolated from raw correspondence output (e.g. in the absence of corrections). Since [10] does not consider motion consistency nor temporal alignment, their depth maps correspond to "raw correspondences" in our method given synchronized input frames.

5 Discussion and Conclusion

We have presented a local spatio-temporal correspondence verification and correction method, and used it to develop a bottom-up solution for video synchronization and dense dynamic modeling. The underlying assumption of local geometric consistency as a guide for spatio-temporal overlap has been proven to be informative across an expanded spatio-temporal scope. We used recent freely available datasets for dynamic 3D reconstruction and these considered a single dynamic element. The multi-body dynamics would be naturally included into our framework as, beyond the attainability of SFM-based camera registration, we only make assumptions on local rigidity and cross-view photo-consistency. Future improvements to our framework include extending the scope of our temporal window through the adoption of robust feature-based tracking frameworks able to sustain and recover tracks across extended periods. Moreover, we will continue to explore more robust structure and synchronization frameworks that leverage our proposed consistency assessment framework as low-level functional building block.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: Proceedings of ICCV (2012)
2. Heinly, J., Schonberger, J., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days *(as captured by the yahoo 100 million image dataset). In: Proceedings of CVPR (2015)
3. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV, pp. 127–134 (2013)
4. Furukawa, Y., Ponce, J.: Towards internet-scale multi-view stereo. In: Proceedings of CVPR 1434 (2010)
5. Joo, H., Park, H., Sheikh, Y.: Map visibility estimation for large scale dynamic 3d reconstruction. In: Proceedings of CVPR (2014)

6. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: Proceedings of ICCV (2015)
7. Kim, H., Sarim, M., Takai, T., Guillemaut, J., Hilton, A.: Dynamic 3d scene reconstruction in outdoor environments. In: Proceedings of 3DPVT (2010)
8. Jiang, H., Liu, H., Tan, P., Zhang, G., Bao, H.: 3d reconstruction of dynamic scenes with multiple handheld cameras. In: Proceedings of ECCV (2012)
9. Taneja, A., Ballan, L., Pollefeys, M.: Modeling dynamic scenes recorded with freely moving cameras. In: Proceedings of ECCV (2010)
10. Mustafa, A., Kim, H., Guillemaut, J., Hilton, A.: General dynamic scene reconstruction from multiple view video. In: Proceedings of ICCV (2015)
11. Oswald, M., Cremers, D.: A convex relaxation approach to space time multi-view 3d reconstruction. In: International Conference on Computer Vision (ICCV) Workshops, pp. 291–298 (2013)
12. Oswald, M., Stühmer, J., Cremers, D.: Generalized connectivity constraints for spatio-temporal 3d reconstruction. In: European Conference on Computer Vision (ECCV) pp. 32–46. IEEE (2014)
13. Djelouah, A., Franco, J.S., Boyer, E., Le Clerc, F., Pérez, P.: Sparse multi-view consistency for object segmentation. Pattern Anal. Mach. Intell. (PAMI) **37**(9), 1890–1903 (2015)
14. Letouzey, A., Boyer, E.: Progressive shape models. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 190–197. IEEE (2012)
15. Wu, C., Varanasi, K., Liu, Y., Seidel, H.P., Theobalt, C.: Shading-based dynamic shape refinement from multi-view video under general illumination. In: International Conference on Computer Vision (ICCV), pp. 1108–1115. IEEE (2011)
16. Guan, L., Franco, J.S., Pollefeys, M.: Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction. Int. J. Comput. Vis. (IJCV) **90**(3), 283–303 (2010)
17. Cagniart, C., Boyer, E., Ilic, S.: Probabilistic deformable surface tracking from multiple videos. In: European Conference on Computer Vision (ECCV), pp. 326–339. Springer, Heidelberg (2010)
18. Russell, C., Yu, R., Agapito, L.: Video pop-up: Monocular 3d reconstruction of dynamic scenes. In: Proceedings of ECCV (2014)
19. Zheng, E., Ji, D., Dunn, E., Frahm, J.: Sparse dynamic 3d reconstruction from unsynchronized videos. In: Proceedings of ICCV, pp. 4435–4443 (2015)
20. Zheng, E., Ji, D., Dunn, E., Frahm, J.: Self-expressive dictionary learning for dynamic 3d reconstruction. arXiv preprint (2014). arXiv:1605.06863
21. Basha, T., Moses, Y., Avidan, S.: Photo sequencing. In: Proceedings of ECCV (2012)
22. Basha, T., Moses, Y., Avidan, S.: Space-time tradeoffs in photo sequencing. In: Proceedings of ICCV (2013)
23. Tuytelaars, T., Gool, L.: Synchronizing video sequences. In: Proceedings of CVPR, pp. 762–768 (2004)
24. Wolf, L., Zomet, A.: Wide baseline matching between unsynchronized video sequences. Int. J. Comput. Vis. **68**, 43–52 (2006)
25. Pundik, D., Moses, Y.: Video synchronization using temporal signals from epipolar lines. In: Proceedings of ECCV, pp. 15–28 (2010)
26. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Proceedings of ECCV (2004)

27. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. Trans. PAMI **34**, 2274 (2012)
28. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: Interactive exploration of casually captured videos. ACM Trans. Graph. **29**, 87 (2010). ACM