Sparse Dynamic 3D Reconstruction from Unsynchronized Videos

Enliang Zheng, Dinghuang Ji, Enrique Dunn, and Jan-Michael Frahm
The University of North Carolina at Chapel Hill
{ezheng,jdh,dunn,jmf}@cs.unc.edu

Abstract

We target the sparse 3D reconstruction of dynamic objects observed by multiple unsynchronized video cameras with unknown temporal overlap. To this end, we develop a framework to recover the unknown structure without sequencing information across video sequences. Our proposed compressed sensing framework poses the estimation of 3D structure as the problem of dictionary learning. Moreover, we define our dictionary as the temporally varying 3D structure, while we define local sequencing information in terms of the sparse coefficients describing a locally linear 3D structural interpolation. Our formulation optimizes a biconvex cost function that leverages a compressed sensing formulation and enforces both structural dependency coherence across video streams, as well as motion smoothness across estimates from common video sources. Experimental results demonstrate the effectiveness of our approach in both synthetic data and captured imagery.

1. Introduction

Scene reconstruction from photo collections has reached a high level of maturity due to the recent progress in structure from motion and stereo estimation [21, 28, 15]. Despite these tremendous advances, these methods only reconstruct the static parts of the environment captured by the photo collections. However, most real-life videos and photos have dynamic elements, e.g., imagery with people as the main object of interest. Take, for example, videos captured at music concerts, sports events, etc. It is these dynamic objects that we often aim to capture as they bring the static scenes to life. The reconstruction of the dynamic objects in these scenes using videos or photos currently falls far behind the maturity for reconstruction of static scene elements. There are some early approaches towards reconstructing dynamic scene elements from ad-hoc capture scenarios [17, 20]. However, there are significant challenges ahead in order to leverage uncontrolled video capture such as those available in crowd sourced video collections.

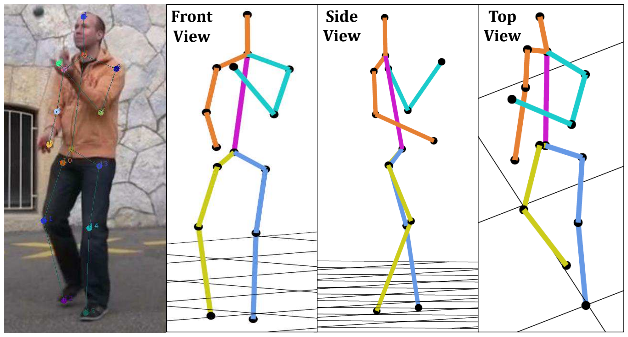

Figure 1: (left) Example frame from the multiple videos capturing a performance serving as input to our method, with overlaid structure (points), and (right three) different views of the reconstructed 3D points. Note that our method only estimates the 3D points but no topology. The skeleton lines are only added for visualization purposes.

In this paper, we specifically target the reconstruction of the shape of dynamic objects captured by a variety of unsynchronized video cameras (See Fig. 2). This setup is encountered, for example, when several people capture an event, such as people dancing, each using their own cameras. In that case, all videos capture the same event, but since the cameras are not synchronized, the temporal order of the frames is only known within each video sequence. Hence, we propose a solution to determining 3D dynamic structure without inter-sequence temporal information, accounting for the potentially different and unknown frame rates of the cameras. Fig. 1 shows one sample output of our approach.

Despite its high relevance to real-life video collections, there are currently no methods that can successfully address this problem. Existing methods for shape reconstruction [19, 23] inherently require temporally ordered image sequences (sequencing information) to reconstruct the 3D points of the dynamic objects. As explained above, with independently captured videos, it is challenging to provide this sequencing information. Zheng *et al*. [29] recently proposed to jointly estimate the photo sequencing and 3D point estimation based on object detections by solving a general-

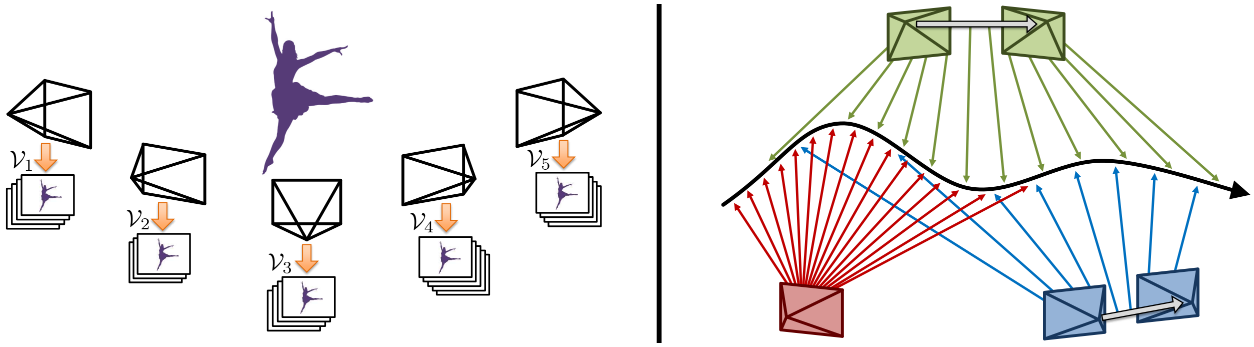

Figure 2: (left) Multiple videos capture a performance, which serves as input to our method, and (right) each input video has a different sampling of a 3D point's trajectory.

ized minimum spanning tree (GMST) problem. However, the GMST problem itself is NP hard, which makes the algorithm not scalable. In this paper, we propose a continuous formulation that jointly poses the problems of dynamic structure estimation and cross-stream image sequencing as a compressive sensing dictionary learning task [11, 1].

The remainder of the paper is organized as follows. We briefly discuss the related work in Section 2. After this discussion, we introduce the foundations of our novel proposed approach in Section 4, followed by a detailed introduction to our model for shape estimation without sequencing in Section 5. Section 6 then describes our proposed efficient optimization solver to minimize the model. We conclude the paper with an experimental evaluation of the proposed approach on real and synthetic data in Section 7.

2. Related Work

Our work is closely related to trajectory triangulation from a monocular image sequence [2, 19, 23, 29, 30]. Avidan and Shashua [2] first coin the task of trajectory triangulation that reconstructs the 3D coordinates of a moving point from monocular images. Their method assumes the dynamic point moves along simple parametric trajectories, such as straight line or conic section. Park *et al.* [19] represent the trajectory with a linear combination of low-order discrete cosine transform (DCT) bases, and the trajectory is triangulated by estimating the coefficients of the linear combination. There are two fundamental limitations of the method as observed in [23]. First, there is no automated scheme to determine the optimal number of DCT bases. Second, the correlation between the object trajectory and the camera motion inherently limits the reconstruction accuracy. Valmadre *et al.* [23] overcomes the first limitation by proposing a new method without using DCT bases. They estimate the trajectory by minimizing the trajectory's response to a bank of high pass filters. To overcome the second limitation, Zhu *et al.* [30] propose to incorporate the 3D structures of a number of key frames to enhance the reconstructability. However, obtaining those key-frame 3D structures requires interaction from human users. The methods in [19, 23, 29, 30] require the sequencing information of the images, but in natural capture setups, the availability of sequencing information and high reconstructability typically cannot be fulfilled simultaneously [30].

Zheng *et al.* [29] address a slightly different problem. They triangulate the object class trajectory, which is defined by the connection of the objects of the same class moving in a common 3D path, from a collection of unordered images. Their method jointly estimates the trajectory and sequencing, but has low scalability and efficiency due to the NP hard GMST problem. In contrast, our proposed method reconstructs the dynamic objects without sequencing information across videos.

One class of related works solve the non-rigid structure from motion (NRSFM) problem that targets simultaneous recovery of the camera motion (rotation) and the 3D structure using image sequence. These methods typically start from a set of 2D correspondences across frames, obtained by optical flow or graph match based matching algorithms [27]. The work by Bregler et al. [8] tackles the NRSFM problem through matrix factorization, with the assumption that deforming nonrigid objects can be represented by a linear combination of low-order shape bases. It was later shown that in order to achieve a unique solution, more than just the orthogonality constraints have to be used [26]. To solve this shape ambiguity, prior knowledge is required to obtain a unique solution. Not until very recently, Dai *et al.* [10] propose a new prior-free method.

As a dual method to above shape-based methods, Akhter *et al.* [16] propose the first trajectory-based NRSFM approach, which leverages DCT bases to approximately represent object point trajectories. While shape-based NRSFM approaches typically do not require the sequencing information, trajectory-based approaches completely fail if image frames are randomly shuffled (as shown in [10]).

At first glance, it seems the shape-based approaches can be applied to our problem without much modification. Nevertheless, these approaches assume orthographic camera model. It has been shown empirically that the extension of these methods to projective camera model is not straightforward [19]. There are approaches for projective non-rigid shape and motion recovery based on tensor estimation [14, 24], but this difficult problem is still under on-going research. Moreover, the NRSFM methods only recover the shape of the object without absolute translation.

Sequencing information is important in trajectory triangulation. Recently, Basha *et al.* [5, 6] target the problem of determining the temporal order of a collection of photos without recovering the 3D structure of the dynamic scene. The method in [5] relies on two images taken from roughly the same location to eliminate the uncertainty in the sequencing. Basha *et al.* [6] later introduce a solution that leverages the known temporal order of the images from each camera. Both of these methods assume dynamic objects move closely to a straight line within a short time period, but in practice points can deviate considerably from the linear motion model. Tuytelaars *et al.* [22] propose a method to automatically synchronize two video sequences of the same event. They do not use any constraints from the scene or cameras, but rather rely on point correspondences among the video sequences.

3. Problem and notations

Let $\{\mathcal{I}\}$ denote an aggregated set of images attained from N video sequences $\{\mathcal{V}_n\}$. Assuming a total of F available images we can denote each individual image as $I_f \in \{\mathcal{I}\}$, where $f = 1, \ldots, F$. Alternatively, we can refer to the m-th frame in the n-th video as $I_{(n,m)} \in \{\mathcal{V}_n\}$, where $n = 1, \ldots, N$ and $m = 1, \ldots, |\{\mathcal{V}_n\}|$.

We now discuss the parametrization of 3D structure within our framework. We assume an *a priori* camera registration through structure-from-motion analysis of static background structures within the environment [25]. Accordingly, for each available image I_f we know the capturing camera's pose matrix $\mathbf{M}_f = [\mathbf{R}_f \mid -\mathbf{R}_f\mathbf{C}_f]$, along with its intrinsic camera matrix $\mathbf{K}_f$.

Without loss of generality, we assume each image I_f captures a common set of P 3D points $\{\mathbf{X}\}$, and assume correspondences of image features $\mathbf{x}_{(f,p)}$, $f \in \{1, \ldots, F\}$ across images are available. Then for each image feature $\mathbf{x}_{(f,p)}$ with $p \in \{1, \ldots, P\}$, we can compute a viewing ray with direction

$$\mathbf{r}_{(f,p)} = \mathbf{R}_f^\top \mathbf{K}_f^{-1} \begin{bmatrix} \mathbf{x}_{(f,p)} \\ 1 \end{bmatrix}.$$

Hence, the position of the dynamic 3D point $\mathbf{X}_{(f,p)}$ corresponding to $\mathbf{x}_{(f,p)}$ can be described by the distance along the ray $\mathbf{r}_{(f,p)}$ given by

$$\mathbf{X}_{(f,p)} = \mathbf{C}_f + d_{(f,p)}\mathbf{r}_{(f,p)}, \tag{1}$$

where $d_{(f,p)}$ is the unknown distance of the 3D point from the camera center.

Given F frames with each frame observing P dynamic 3D points, we denote our aggregated observed 3D datum as

$$\mathbb{X} = \begin{bmatrix} \mathbf{X}_{(1,1)} & \cdots & \mathbf{X}_{(1,F)} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{(P,1)} & \cdots & \mathbf{X}_{(P,F)} \end{bmatrix} = [\mathbf{S}_1 \ \cdots \ \mathbf{S}_F] \tag{2}$$

where the f-th column of the matrix $\mathbb{X}$, denoted as $\mathbf{S}_f$, is obtained by stacking all the P 3D points observed in the f-th frame. Since the point $\mathbf{X}_{(f,p)}$ has one unknown variable $d_{(f,p)}$, we denote the matrix describing each of the dependent variables associated with $\mathbb{X}$ as

$$\mathbf{d} = \begin{bmatrix} d_{(1,1)} & \cdots & d_{(1,F)} \\ \vdots & \ddots & \vdots \\ d_{(P,1)} & \cdots & d_{(P,F)} \end{bmatrix}. \tag{3}$$

The use of $\mathbf{d}$ within our framework will be explained in section 6. Our task is to recover $\mathbb{X}$ from the 2D measures without image sequencing information across videos.

4. Principle

We first describe the observations motivating our solution before presenting a detailed description of our proposed method for determining 3D structure from a set of unsynchronized videos.

For our method, we assume a smooth 3D motion under the sampling provided by the videos. Hence, we can approximate the observed 3D structure $\mathbf{S}_f$ observed in image f in terms of a linear combination of the structures corresponding to the set of immediately preceding ($\mathbf{S}_{prev}$) and succeeding ($\mathbf{S}_{next}$) frames in time. In this way, we have

$$\mathbf{S}_f \approx t \cdot \mathbf{S}_{prev} + (1-t) \cdot \mathbf{S}_{next}, \tag{4}$$

for $0 \leq t \leq 1$. If our structure matrix $\mathbb{X}$ from Equation (2) was temporally ordered, which it is not in general, the two neighboring frames would be $\mathbf{S}_{f-1}$ and $\mathbf{S}_{f+1}$. Clearly, such perfect temporal order can be extracted from a single video sequence. However, the reconstructibility constraints [23, 29] make single camera structure estimation ill-posed. Hence, *we rely on inter-sequence temporal ordering information to solve the 3D trajectory triangulation problem.* The absence of a global temporal ordering requires us to search for temporal adjacency relations across the different video streams of potentially different frame rates.

In the most simple scenario, the pool of candidate neighboring frames is comprised by all other frames except f. Writing the 3D points of the current frame $\mathbf{S}_f$ as a linear combination of other frames, we have

$$\mathbf{S}_f = \mathbb{X}\mathbf{T}_f, \tag{5}$$

where $\mathbf{T}_f = \left(t_{(f,1)}, \ldots, t_{(f,f-1)}, 0, t_{(f,f+1)}, \ldots, t_{(f,F)}\right)^\top$ is a vector of length F representing the coefficients for the linear combination. Note that the f-th element in $\mathbf{T}_f$ equals 0, since the f-th column of $\mathbb{X}$ (corresponding to $\mathbf{S}_f$) is not used as an element of the linear combination. Moreover, since only the structure estimates $\mathbf{S}_j$ in the close temporal neighborhood of $\mathbf{S}_f$ are likely to provide a good approximation, we expect the vector $\mathbf{T}_f$ to be sparse. Accordingly, we propose to find the local temporal neighborhood of a 3D point set $\mathbf{S}_j$ through a compressive sensing formulation by introducing the l_1 norm as follows,

$$\underset{\mathbf{T}_f}{\text{minimize}} \; ||\mathbf{S}_f - \mathbb{X}\mathbf{T}_f||_2^2 + \lambda||\mathbf{T}_f||_1, \tag{6}$$

where λ is a positive weight. Here, the l_1 norm serves as an approximation of the l_0 norm and favors the attainment of sparse coefficient vectors $\mathbf{T}_f$ [3].

Moreover, we incorporate the desired properties of our linear combination framework (Eq. (4)) and reformulate Eq. (6) as

$$\begin{aligned} \underset{\mathbf{T}_f}{\text{minimize}} \quad & ||\mathbf{S}_f - \mathbb{X}\mathbf{T}_f||_2^2 \\ \text{subject to} \quad & \mathbf{T}_f \cdot \mathbf{1}_{F\times 1} = 1 \\ & \mathbf{T}_f \geq 0 \quad \forall f \in \{1, \ldots, F\}. \end{aligned} \tag{7}$$

The affine constraints of Eq.(7) constrain the variable $\mathbf{T}_f$ to reside in the simplex Δ_f defined as

$$\Delta_f \triangleq \{\mathbf{T}_f \in \mathbb{R}^F \text{ s.t. } \mathbf{T}_f \geq 0, t_{(f,f)} = 0 \text{ and } \sum_{j=1}^{F} t_{(f,j)} = 1\} \tag{8}$$

Eq. (7) is a variant of compressive sensing that still keeps the sparsity-inducing effect [3, 9]. For the considered problem of sequencing with known structure, we know that the sparsity can be achieved at the point that satisfies the simplicial constraint. A similar formulation has been used in modeling archetypal analysis for representation learning [9]. They also provide a new efficient solver for this kind of problem.

Finally, we generalize our formulation from Eqs. (6) and (7) to include all available structure estimates $\mathbf{S}_f$, with $f = 1, \ldots, F$ into the following equation

$$\begin{aligned} \underset{\mathbb{T}}{\text{minimize}} \quad & ||\mathbb{X} - \mathbb{X}\mathbb{T}||_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{T}_f \in \Delta_f, f = 1, \cdots, F \end{aligned} \tag{9}$$

where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm, and $\mathbb{T} = [\mathbf{T}_1 \ \ldots \ \mathbf{T}_F]$ is an $F \times F$ matrix with the f-th column equal to $\mathbf{T}_f$. By construction, the matrix $\mathbb{T}$ has all the diagonal elements equal to zero.

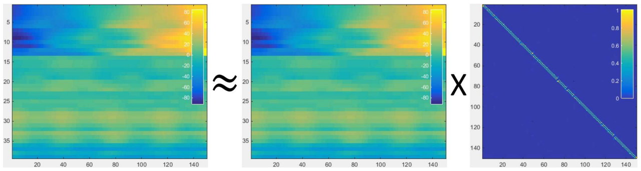

Figure 3: We illustrate the output of Eq. (9) on a real motion capture data of 13 3D points over 150 frames. Each element in $\mathbb{X}$ corresponds to a ground truth coordinate value. The estimation of $\mathbb{T}$ through l_1 compressive sensing approximates the correct ordering after enforcing all elements in the diagonal to be equal to 0.

As an illustration of the validity of our compressed sensing formulation, Fig. 3 shows the output of Eq. (9) on a "faint" real motion capture dataset presented in [19] given the known 3D points $\mathbb{X}$. Although image sequencing is assumed unknown, we show results in temporal order for visualization purposes. The coefficients in $\mathbb{T}$ approximate a matrix having non-vanishing values only on the locations directly above and below the main diagonal. This indicates that, in this specific example, the 3D points $\mathbf{S}_f$ are a linear combination of $\mathbf{S}_{f-1}$ and $\mathbf{S}_{f+1}$.

Minimizing Eq. (7) is equivalent to finding most related shapes to linearly represent $\mathbf{S}_f$. It is usually true that the temporally close shapes $\mathbf{S}_{f-1}$ and $\mathbf{S}_f$ are most related, and therefore it is able to reveal the local temporal information based on the non-vanishing values in $\mathbb{X}$. However, if object motion is repetitive or that the object is static for a period of time, there is no guarantee that the most related shapes are temporally close. Even though this is true, it does not cause any problem for our method aiming at 3D reconstruction.

To validate our prior of sparse representation for real motion, we quantitatively evaluate the estimated coefficients $\mathbb{T}$ by minimizing Eq. (9), on all the three real motion capture datasets presented in [19]. For a shape at a given time sample, we measure the sum of the two largest estimated coefficient values for this sample, and the frequency with which these top two coefficients correspond to the ground truth temporally neighboring shape samples. Given our prior, values of 1 for both measures are expected. The average values we get are 0.9860 and 0.9895, supporting the validity of our prior.

We note that the self-representation in Eq. (9) is previously used in sparse subspace clustering [12], where the element in each subspace can be sparsely represented by other elements in the same subspace, and the coefficients of sparse coding is used to build a graph for clustering.

5. Method

We address the problem of estimating sparse dynamic 3D structure from a set of spatially registered video sequences with unknown temporal overlap. Section 4 presented a compressive sensing formulation leveraging the self-expressiveness of all the shapes in the context of known 3D geometry. However, our goal is to estimate the unknown structure without sequencing information. To this end, we define our dictionary as the temporally varying 3D structure, and propose a compressive sensing framework which poses the estimation of 3D structure as a dictionary learning problem. We solve this problem in an iterative and alternating manner, where we optimize for 3D structure while fixing the sparse coefficients, and *vice versa*. This is achieved through the optimization of a biconvex cost function that leverages the compressed sensing formulation described in Section 4 and, additionally, enforces both structural dependency coherence across video streams, as well as motion smoothness among estimates from common video sources.

5.1. Cost function

To achieve the stable estimation of both the structure $\mathbb{X}$ and the sequencing information $\mathbb{T}$, we extend our formulation from Equation (9) to the following cost function,

$$\begin{aligned} \underset{\mathbb{X},\mathbb{T}}{\text{minimize}} \quad & ||\mathbb{X} - \mathbb{X}\mathbb{T}||_F^2 + \lambda_1 \Psi_1(\mathbb{T}) + \lambda_2 \Psi_2(\mathbb{X}) \\ \text{subject to} \quad & \mathbf{T}_f \in \Delta_f, f = 1, \cdots, F; \end{aligned} \tag{10}$$

where $\Psi_1(\mathbb{T})$ and $\Psi_2(\mathbb{X})$ are two convex cost terms regulating the spatial relationships between 3D observations within and across video image streams.

5.2. Coefficient relationships

As described in Section 4, a given structure $\mathbf{S}_f$ in frame f can be attained from the linear combination of the 3D points $\mathbf{S}_i$ captured in other frames. The coefficients or weights of the linear combination are given by the elements of the matrix $\mathbb{T}$. In particular, the element on the f-th column and j-th row of $\mathbb{T}$ is denoted as $t_{(f,j)}$, and it indicates that in order to estimate the 3D points in $\mathbf{S}_f$, the relative contribution (weight) from $\mathbf{S}_j$ is equal to the magnitude of $t_{(f,j)}$. Similarly, $t_{(j,f)}$ represents the contribution of $\mathbf{S}_j$ towards the 3D points in $\mathbf{S}_f$. Accordingly, a value of $t_{(j,k)} = 0$ indicates the absence of any contribution from $\mathbf{S}_k$ to $\mathbf{S}_j$, which is desired for tempo/spatially non-proximal 3D shapes.

We note that, if $\mathbf{S}_f$ contributes to $\mathbf{S}_j$, it means the two sets of points are highly related, which further implies that $\mathbf{S}_j$ should reciprocally contribute to estimating $\mathbf{S}_f$. We deem this reciprocal influence within our estimation process as *structural dependence coherence* and develop a cost term that contributes to enforce this property within the estimation of $\mathbb{T}$. We encode this relationship into our cost function as an additional term of the form

$$\Psi_1(\mathbb{T}) = ||\mathbb{T} - \mathbb{T}^\top||_F^2 \tag{11}$$

A strict interpretation of the above formulation aims to identify symmetric matrices. In general, the reciprocal influence between $\mathbf{S}_f$ and $\mathbf{S}_j$ does not imply symmetric contribution, as the values of $t_{(f,j)}$ and $t_{(j,f)}$ depend on the actual 3D motion being observed. More specifically, these values describe the linear structural dependencies between two different, but overlapping, 3-tuples of 3D points, e.g. ($\mathbf{S}_i$,$\mathbf{S}_f$,$\mathbf{S}_j$) and ($\mathbf{S}_f$,$\mathbf{S}_j$,$\mathbf{S}_k$) as illustrated in Fig. 4. Following the example of Fig. 4 it can be seen that $\mathbf{S}_i$ and $\mathbf{S}_j$ are at equal distance to $\mathbf{S}_f$ and hence equally contribute to it, i.e. $t_{(f,j)} = \frac{1}{2}$. However, in order to determine the linear combination weights for specifying $\mathbf{S}_j$, we need to consider $\mathbf{S}_f$ and $\mathbf{S}_k$. Given their distances to $\mathbf{S}_j$ of 0.5 and 0.1 respectively the weight of $t_{(j,f)} = \frac{1}{6}$, which is significantly lower than $t_{(f,j)}$. Accordingly, we do not expect a fully symmetric weight matrix $\mathbb{T}$. However, given our expectation of a sparse coefficient matrix $\mathbb{T}$, we can focus on finding congruence between the zero-value elements of the $\mathbb{T}$ and $\mathbb{T}^\top$, which $\Psi_1(\mathbb{T})$ effectively encodes. Moreover, $\Psi_1(\mathbb{T})$ is convex, which enables its deployment within our biconvex optimization framework.

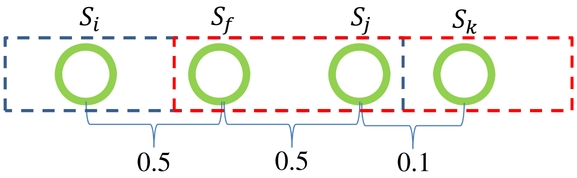

Figure 4: Illustration of the triplets influencing the weights for $\mathbf{S}_f$ and $\mathbf{S}_j$ leading to an asymmetric $\mathbb{T}$. The values in the figure are the distance between points.

5.3. Sequencing information

As mentioned in Section 4, while the availability of video sequences enables enforcing constraints among frames attained from the same video, these constraints are insufficient to robustly estimate 3D geometry. Under the assumption of sufficiently smooth 3D motion w.r.t. the framerate of each video capture, we define a 3D spatial smoothness term that penalizes large displacements among successive frames from the same video. Therefore, we define a pairwise term over the values of $\mathbb{X}$

$$\Psi_2(\mathbb{X}) = \sum_{n=1}^{N} \sum_{m=1}^{|\mathcal{V}_n|-1} \left|\left|\mathbf{X}_{(n,m)} - \mathbf{X}_{(n,m+1)}\right|\right|_2^2 \tag{12}$$

where n is the video index, m is the image index within a video and $|\mathcal{V}_n|$ denotes the number of video frames within each sequence. Note that $\Psi_2(\mathbb{X})$ does not explicitly enforce ordering information across video sequences, but instead fosters a compact 3D motion path within a sequence. Moreover, $\Psi_2(\mathbb{X})$ is a convex term.

However, this regularization term $\Psi_2(\mathbb{X})$ is a double-edged sword. Since this term minimizes the sum of squared distances, and if a video camera is static or has small motion, the estimated 3D points are likely to be pulled towards the camera center. This typically biases the estimated 3D points slightly away from their real positions. Therefore, we propose to first minimize Eq. (10) to obtain values for $\mathbb{X}$ and $\mathbb{T}$, and then taking those values as initialization, we further optimize the problem with weight of $\Psi_2(\mathbb{X})$ (*i.e.* λ_2) set to 0.

5.4. Dictionary space reduction

The first cost term in Eq. (10) functions as searching shapes in the dictionary to sparsely represent each shape. The searching space can be reduced if some elements of $\mathbb{T}$ are forced to be 0. As mentioned, the diagonal elements of $\mathbb{T}$ are forced to be 0, because a shape is not used to represent its own. Moreover, it is possible that if *a priori* knowledge of rough temporal information across video steams is available, we can use it to reduce the searching space.

In our formulation, we explicitly enforce that the shape observed by one video is not used to represent the shape observed in the same video, because the reconstructibility analysis in [23, 29] shows such estimation is ill-posed. In our implementation, this is achieved by not defining the corresponding variables in $\mathbb{T}$ during the optimization.

6. Optimization

The biconvex function in Eq. (10) is non-convex, but it is convex if one set of the variables $\mathbb{X}$ or $\mathbb{T}$ is fixed. To optimize Eq. (10), though more complicated dictionary update scheme such as K-SVD [1] is possible, in this paper we use the simplest optimization scheme that alternates the optimizations over $\mathbb{X}$ and $\mathbb{T}$. Since the alternating optimization steps need to be performed until convergence, it requires each step to be reasonably fast. Although optimizing over $\mathbb{X}$ is relatively easy, optimizing over $\mathbb{T}$ is relatively more difficult due to the simplicial constraint. We find that optimizing over $\mathbb{T}$ with a general solver, such as CVX [13], is too slow for moderate number of total frames F. Moreover, during our iterative optimization, the output of the previous step can be fed into the current step as a good initializaiton (hot start), but typically the general solver does not allow for a hot start. To solve the problem of speed and scalability, we propose a new solver based on alternating direction method of multipliers (ADMM) [7].

6.1. Optimize over $\mathbb{X}$

If $\mathbb{T}$ in Eq. (10) is fixed, the optimization over $\mathbb{X}$ is straightforward. After substituting Eq. (1) into Eq. (10), we get a quadratic programming problem without any constraint on the unknown variable $\mathbf{d}$. The solution can be found at zero value of the derivative of the cost function over $\mathbf{d}$.

6.2. Optimize over $\mathbb{T}$

The optimization over $\mathbb{T}$ is more complex mainly due to the simplex constrains. By fixing the variable $\mathbb{X}$ in Eq. (10), the cost function becomes,

$$\begin{aligned} \underset{\mathbb{T}}{\text{minimize}} \quad & ||\mathbb{X} - \mathbb{X}\mathbb{T}||_{\text{F}}^2 + \lambda_1 ||\mathbb{T} - \mathbb{T}^\top||_2^2 \\ \text{subject to} \quad & \mathbf{T}_f \in \Delta_f, f = 1, \cdots, F \end{aligned} \tag{13}$$

Notice that if the term $||\mathbb{T} - \mathbb{T}^\top||_{\text{F}}^2$ vanishes, the cost function is the same to Eq. (9). Eq. (9) can be decomposed into Eq. (7), and optimized over $\mathbf{T}_f$ for each $f = 1, \ldots, F$ independently. Therefore the number of variables for each subproblem is much smaller comparing to the total number of variables in $\mathbb{T}$, and it can be parallelized on the level of subproblems. Moreover, Chen *et al.* [9] propose a fast solver to the optimization problem in Eq. (7) based on an active-set algorithm that can benefit from the solution sparsity. However, the cost term $||\mathbb{T} - \mathbb{T}^\top||_{\text{F}}^2$ prevents the decomposition.

In this paper, we propose an ADMM algorithm that enables the decomposition. By introducing a new auxiliary variable $\mathbb{Z}$, Eq. (13) can be rewritten as

$$\begin{aligned} \underset{\mathbb{T}}{\text{minimize}} \quad & ||\mathbb{X} - \mathbb{X}\mathbb{T}||_{\text{F}}^2 + \lambda_1 ||\mathbb{Z} - \mathbb{Z}^\top||_{\text{F}}^2 \\ \text{subject to} \quad & \mathbf{T}_f \in \Delta_f, f = 1, \cdots, F \\ & \mathbb{T} = \mathbb{Z} \end{aligned} \tag{14}$$

The resulting ADMM algorithm iterates until convergence. For each iteration, there are three steps. In step 1

$$\begin{aligned} \mathbb{T}^{k+1} = \underset{\mathbf{T}_f \in \Delta_f, \, \text{for} 1 \leq f \leq F}{\text{argmin}} & ||\mathbb{X} - \mathbb{X}\mathbb{T}||_{\text{F}}^2 \\ & + \text{vec}(\mathbb{Y}^k)^\top \text{vec}(\mathbb{T}) + \frac{\rho}{2} ||\mathbb{T} - \mathbb{Z}^k||_{\text{F}}^2 \end{aligned} \tag{15}$$

where the superscript k is the iteration index. $\mathbb{Y}^k$ is the matrix of dual variables and is initialized with 0. Note that the values of $\mathbb{Y}^k$ and $\mathbb{Z}^k$ are known during this step, we only optimize over the variable $\mathbb{T}$. The optimization can be decomposed into optimizing over $\mathbf{T}_f$ independently and in parallel, and minimized by the fast solver proposed in [9]. In step 2

$$\begin{aligned} \mathbb{Z}^{k+1} = & \underset{\mathbb{Z}}{\text{argmin}} \, \lambda_1 ||\mathbb{Z} - \mathbb{Z}^\top||_{\text{F}}^2 - \text{vec}(\mathbb{Y}^k)^\top \text{vec}(\mathbb{Z}) \\ & + \frac{\rho}{2} ||\mathbb{T}^{k+1} - \mathbb{Z}||_{\text{F}}^2 \end{aligned} \tag{16}$$

This is a quadratic programming in the unknown variable $\mathbb{Z}$ without constraint, and can be easily solved by setting the derivative of Eq. (16) with respect to $\mathbb{Z}$ equal to 0. In step 3, the dual variables $\mathbb{Y}$ are updated directly according to

$$\mathbb{Y}^{k+1} = \mathbb{Y}^k + \rho(\mathbb{T}^{k+1} - \mathbb{Z}^{k+1}) \tag{17}$$

The three Eqs. (15), (16) and (17) iterates until the stop criterion is met. We use the stop criterion proposed in [7].

6.3. Initialization of the Optimization

Given the non-convexity of our original cost function Eq. (10), the accuracy of our estimates is sensitive to the initialization values used by our iterative optimization. Hence, we designed a 3D structure (i.e. $\mathbb{X}$) initialization mechanism aimed at enhancing the robustness and accelerating the convergence of our biconvex framework. While our approach explicitly encodes the absence of concurrent 3D observations, we aim to leverage the existence of nearly-incident corresponding viewing rays as a cue for the depth initialization of a given 3D point $\mathbf{X}_{(f,p)}$. To this end, we identify for each bundle of viewing rays captured in I_f, (i.e. associated with a given shape structure $\mathbf{S}_f$) an alternative structure instance captured at I_j that minimizes the Euclidean 3D triangulation error across all corresponding viewing rays. In oder to avoid a trivial solution arising from the small-baseline typically associated with consecutive frames of single video, we restrict our search to ray bundles captured from distinct video sequences.

The position of each point $\mathbf{X}_{f,p}$ in $\mathbf{S}_f$ is determined by $d_{(f,p)}$ as in Eq. (1). Denoting $\mathbf{d}_f = [d_{(f,1)}, \ldots, d_{(f,P)}]$, we can find the distance between shape of $\mathbf{S}_f$ and $\mathbf{S}_j$ by minimizing the following cost function over the unknown variables $\mathbf{d}_f$ and $\mathbf{d}_j$

$$\{\mathbf{d}_f^*, \mathbf{d}_j^*\} = \underset{\mathbf{d}_f, \mathbf{d}_j}{\text{argmin}} ||\mathbf{S}_f - \mathbf{S}_j||_2^2 \tag{18}$$

This is a quadratic cost function with a closed-form solution corresponding to the intersection points of each corresponding pair of viewing rays and their common normal. We then build a distance matrix $\mathbf{D}$ with element $D_{(f,j)}$ equal to the minimized Euclidean distance value of Eq. (18). If the frames f and j are from the same video, $D_{(f,j)}$ is set to infinity. Next, we find many pseudo-intersection points with negative ray depth for spatio-temporally distant pairs of viewing rays, and the corresponding element in $\mathbf{D}$ is also set to infinity. Finally, we determine the minimum element of each f-th row in our distance matrix $\mathbf{D}$ and assign the corresponding depth values $\mathbf{d}_f^*$ as our initialization for the definition of our 3D structure $\mathbf{X}_f$.

7. Experiments

We evaluate our algorithm on both synthetic and real data. In our experiments, the values λ_1 and λ_2 in Eq.(10) are set empirically to 1 and 0.1 for all the experiments.

7.1. Synthetic data

To generate synthetic data, we use the motion-capture datasets “faint”, “walk”, and “stand” from [18], and leverage them as ground truth structure for our estimation. The datasets are comprised of the temporal sequences of a common set of 3D points, which correspond within our framework as ground truth structure $\mathbb{X}_{GT}$. These 3D points are projected onto virtual cameras to generate input 2D measures into our methods. $\mathbb{X}_{GT}$. We select the virtual camera to have a resolution of 1M and focal length of 1000, and positioned three static cameras at a common distance from the centroid defined by $\mathbb{X}_{GT}$. The ratio between the distance to the centroid and the maximal distance between any two points in $\mathbb{X}_{GT}$ is set to one. Every third temporal 3D capture is assigned to each camera to build three disjoint image sequences. To test the robustness, we add zero-mean Gaussian noise to the 2D measures with different standard deviations. The accuracy is defined as the mean error of each 3D point from the ground truth. We report the accuracy of our 3D structure estimation using the initialization mechanism proposed in Section 6.2. Figure 5 illustrates the overall accuracy over the evaluated datasets.

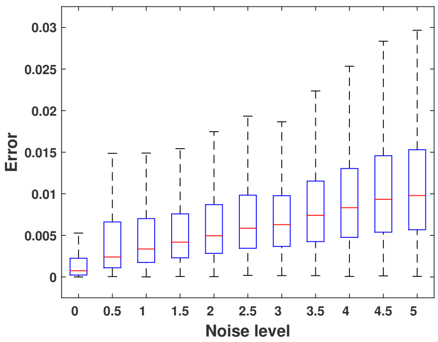

Figure 5: The error distributions at different noise levels. The noise level is defined as the standard deviation of zero-mean Gaussian noise in pixels.

We also quantitatively evaluate the estimated $\mathbb{T}$, obtained by minimizing Eq. (10). Using the same two measures described in Sec. 4, we get values of 0.9797 and 0.9881, which shows our formulation is valid.

7.2. Real datasets

For experiments using real image capture we use the Juggler and Rothman datasets from [4]. We do not use the datasets in [5, 19] because they only provide images with large temporal discrepancy, and therefore each shape cannot be well approximated by the other shapes (*i.e.* Eq. (4) does not hold). We perform manual feature labeling on the input sequences and provide the attained set of 2D measurements as input for our estimation process. We also capture a new dataset of a person juggling (called Juggler2) by three iPhone6 (downsampled to 10 Hz) and one iPhone5 (downsampled to 6.25 Hz) with no temporal synchronization. When testing our algorithm on this dataset, we also include the juggling balls as feature points for reconstruction. For visualization purposes, Fig. 6 depicts the estimated 3D geometry by connecting the estimated position of the detected joint elements through 3D line segments.

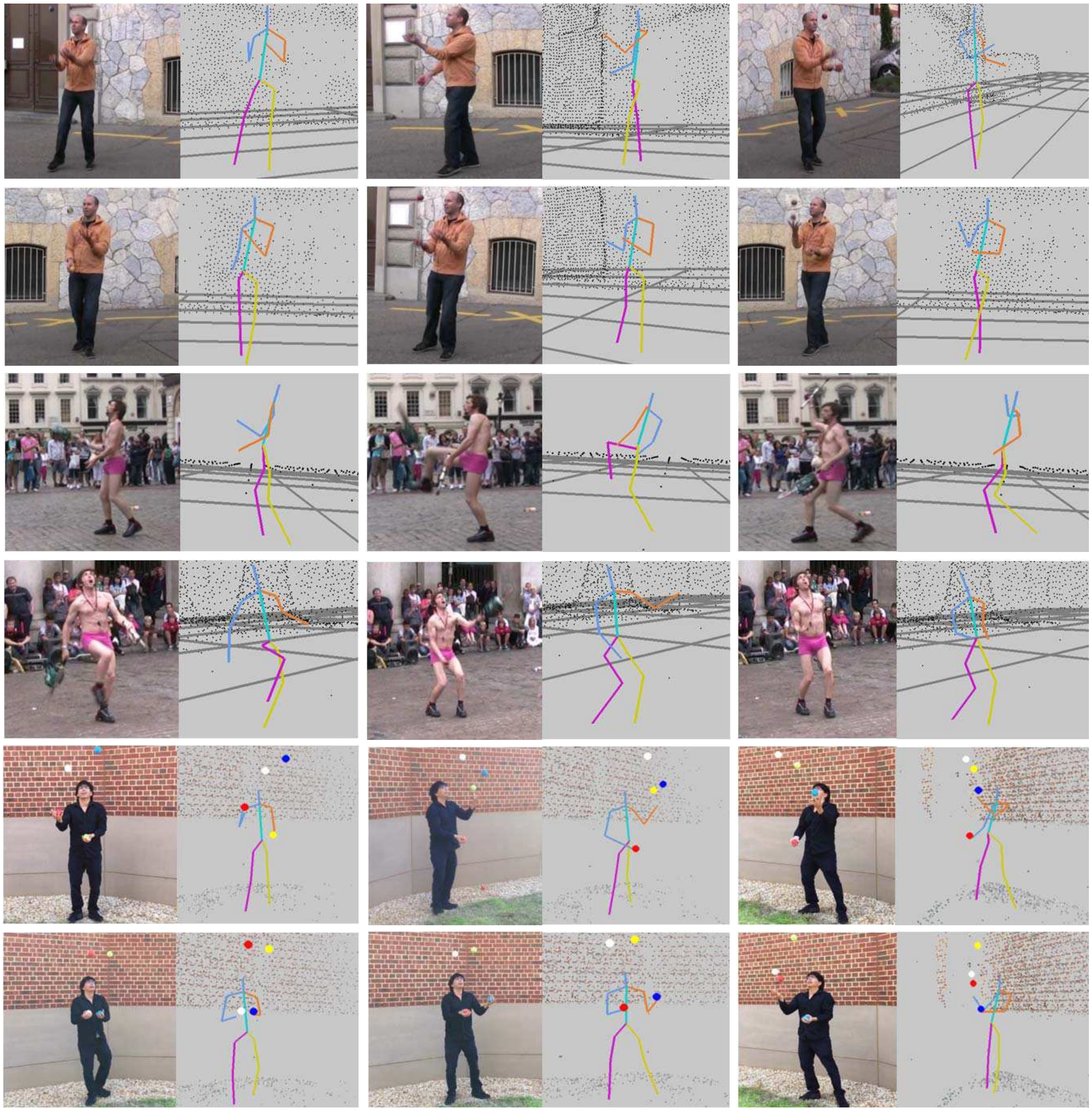

Figure 6: Example results using the real datasets Juggler (150 images, 3 videos) and Rothman (100 images, 2 videos) from [4] and Juggler2 (212 images, 4 videos). All the datasets are captured by handheld cameras.

8. Conclusion

The contributions of our framework encompass:

1. **Problem Definition**. We are the first to address the problem of dynamic 3D structure estimation using unsynchronized cross-video streams.
2. **Methodology Formulation**. We pose the problem in terms of a dictionary learning and compressive encoding framework leveraging a novel data-adaptive local 3D interpolation model.
3. **Implementation Mechanisms**. We define and solve a biconvex optimization problem and develop an efficient ADMM-based solver amenable for parallel implementation.

Our proposed method was successfully evaluated on real and synthetic data. It is a first step towards dynamic 3D modeling in the wild.

Acknowledgment. This research was supported by NSF grants IIS-1349074 and CNS-1405847.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. Svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 2006.

[2] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 2012.

[4] L. Ballan, G. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. In *ACM Transactions on Graphics (TOG)*, 2010.

[5] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. *ECCV*, 2012.

[6] T. Basha, Y. Moses, and S. Avidan. Space-time tradeoffs in photo sequencing. *ICCV*, 2013.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[8] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. 2000.

[9] Y. Chen, J. Mairal, and Z. Harchaoui. Fast and Robust Archetypal Analysis for Representation Learning. In *CVPR*, 2014.

[10] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 2006.

[12] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009.

[13] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[14] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*. 2008.

[15] J. Heinly, J. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). *CVPR*, 2015.

[16] A. Ijaz, S. Yaser, K. Sohaib, and K. Takeo. Nonrigid structure from motion in trajectory space. In *NIPS*, 2008.

[17] D. Ji, E. Dunn, and J. Frahm. 3D Reconstruction of Dynamic Textures in Crowd Sourced Data. *ECCV*, 2014.

[18] H. S. Park and Y. Sheikh. 3D reconstruction of a smooth articulated trajectory from a monocular image sequence. In *ICCV*, 2011.

[19] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *ECCV 2010*. 2010.

[20] C. Russell, R. Yu, and L. Agapito. Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes. In *ECCV*. 2014.

[21] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 2006.

[22] T. Tuytelaars and L. V. Gool. Synchronizing video sequences. In *Computer Vision and Pattern Recognition*, 2004.

[23] J. Valmadre and S. Lucey. General trajectory prior for non-rigid reconstruction. In *CVPR*, 2012.

[24] R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *ECCV 2006*. 2006.

[25] C. Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision*, 2013.

[26] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *ECCV*, 2004.

[27] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *TPAMI*, 2016.

[28] E. Zheng, E. Dunn, V. Jojic, and J. Frahm. Patchmatch based joint view selection and depthmap estimation. In *CVPR*, 2014.

[29] E. Zheng, K. Wang, E. Dunn, and J. Frahm. Joint Object Class Sequencing and Trajectory Triangulation (JOST). In *ECCV*. 2014.

[30] Y. Zhu, M. Cox, and S. Lucey. 3D motion reconstruction for real-world camera motion. In *CVPR*, 2011.