Self-expressive Dictionary Learning for Dynamic 3D Reconstruction

Enliang Zheng, *Member, IEEE,* Dinghuang Ji, Enrique Dunn, *Member, IEEE,*
and Jan-Michael Frahm, *Member, IEEE*

Abstract—We target the problem of 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, where the dictionary is defined as an aggregation of the temporally varying 3D structures. Given the smooth motion of dynamic objects, we observe any element in the dictionary can be well approximated by a sparse linear combination of other elements in the same dictionary (*i.e.* self-expression). 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. We further analyze the reconstructability of our approach under different capture scenarios, and its comparison and relation to existing methods. Experimental results on large amounts of synthetic data as well as real imagery demonstrate the effectiveness of our approach.

Index Terms—Dictionary learning, self-expression, unsynchronized videos, dynamic 3D reconstruction.

1 Introduction

Thanks to the rapid development of mobile technology, it has become common that many people use their own cameras to capture a common event of interest, such as a concert or a wedding. These real-life videos and photos usually have the dynamic objects as the main focus of the scene. With the bursting growth of such crowd-sourced data, it is of interest to develop methods of dynamic scene analysis that enrich understanding and visualization of the captured events.

In this work, we target the problem of dynamic 3D object reconstruction from multiple unsynchronized videos. More specifically, the method takes as input a collection of video streams without inter-sequence temporal information. The video streams could potentially have different, irregular, and unknown frame rates (see Fig. 2). As output, the method reconstructs the 3D positions of sparse feature points at each time instance (*e.g.*, Fig. 1). Dynamic object reconstruction from unsynchronized videos is a challenging problem due to various factors, such as unknown temporal overlap among video streams, possible non-concurrent captures, and dynamic object motion. Any of these factors impedes the valid reconstruction from traditional 3D triangulation, which relies on the assumption of concurrent captures or a static scene.

Despite the ubiquity of uncontrolled video collections, there are currently no methods that can successfully address our problem. Static scene reconstruction from photo collections has reached a high level of maturity [2], [3] thanks to the development of structure from motion and depth estimation, but the reconstruction of dynamic objects using videos currently falls far behind the maturity of reconstruction of static scene elements. Existing methods of trajectory triangulation [4], [5] from monocular image sequences inherently require temporal order information (sequencing information). However, with independently captured videos, it is challenging to obtain this information across videos. Zheng *et al.* [6] recently propose to jointly estimate the photo sequencing and 3D point by solving a generalized minimum spanning tree (GMST) problem. However, the NP-hard GMST problem itself limits the scalability of the approach. Also in this vein, the non-rigid structure from motion (NRSFM) problems have received extensive study over the last two decades [7], [8], [9], [10], [11], but such methods are still under further exploration, especially if a perspective camera model is applied.

To solve the problem, we observe that, given the smooth

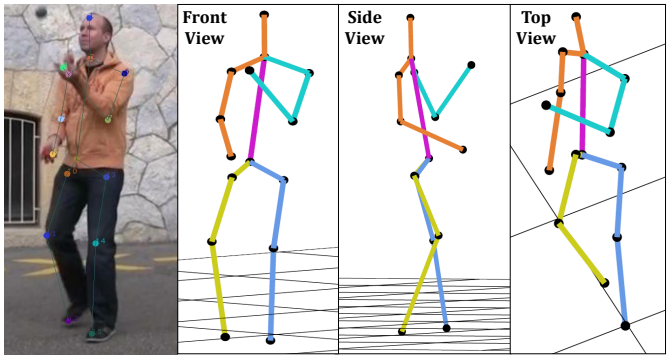


Fig. 1: (Left) Example frame from the multiple videos capturing a performance serving as input to our method, with overlaid structure (points), and (right three) different views of the reconstructed 3D points. Note our method only estimates the 3D points but no topology. The skeleton lines are plotted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

- *Enliang Zheng, Dinghuang Ji and Jan-Michael are with the Department of Computer Science, the University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599. E-mail: {ezheng, jdh, jmf}@cs.unc.edu.*
  Enrique Dunn is with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, 07030. E-mail: edunn@stevens.edu.
- *An earlier version of this paper appears in the International Conference on Computer vision (ICCV), Dec. 13-16, 2015 [1].*

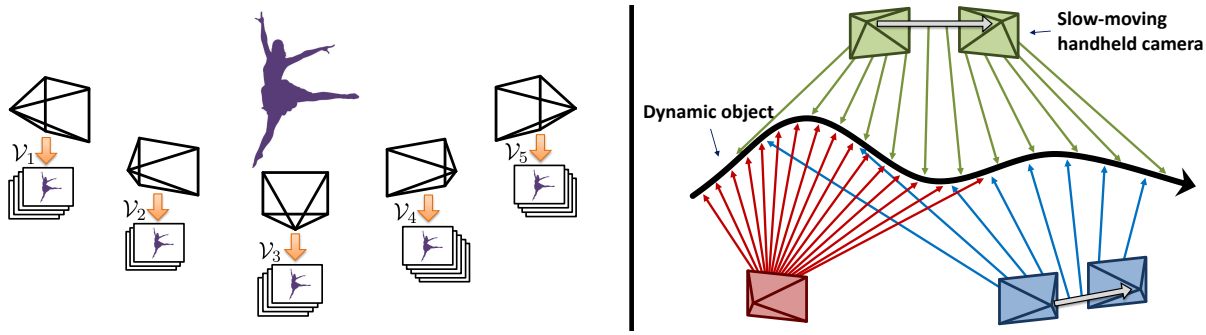


Fig. 2: Left: Multiple videos capture a performance. The corresponding set of independent image streams serves as input to our method. Right: Each input video has a different sampling of a 3D point's trajectory.

motion of a dynamic object, any 3D shape at one time instance can be sparsely approximated by other shapes across time. Based on this self-expressive representation, our solution leverages the compressive sensing technique (l_1 norm), and tackles the problem in a dictionary learning framework [12], [13], where the dictionary is defined by the temporally varying 3D structure. Though the self-expression technique has been previously used in subspace clustering for motion segmentation [14], and dictionary learning has been used in other applications such as image denoising [13] and human pose estimation from single image [15], we are the first to explore learning a self-expressive dictionary for the problem of dynamic object reconstruction from unsynchronized videos.

The remainder of the paper is organized as follows. We briefly discuss related works in Sec. 2. After introducing the notations in Sec. 3, we begin describing foundations of our proposed approach in Sec. 4. Sec. 5 presents our model for dynamic object reconstruction, followed by the parameterization of the 3D structure given different kinds of 2D measures in Sec. 6. Sec. 7 describes our ADMM-based optimization solver to minimize the model. Then, Sec. 8 illustrates the reconstructablity of our algorithm. We provide experimental evaluations in Sec. 9, and conclude the paper in Sec. 10

2 Related Work

2.1 Trajectory triangulation

Our work is closely related to trajectory triangulation from monocular images [4], [5], [6], [16], [17], [18]. Avidan and Shashua [16] first coined the task of trajectory triangulation that reconstructs the 3D coordinates of a moving point from monocular images. That is, each dynamic point is observed by only one camera at a time. Their method assumes the dynamic point moves along a simple parametric trajectory, such as a straight line or a conic section.

Recent works focus on a more general model for trajectory triangulation. Park *et al.* [4] 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 their method as observed in [5]. First, there is no automated scheme to determine the optimal number (K) of DCT bases. Second, the correlation between the object trajectory and the camera motion inherently limits the reconstruction accuracy. To overcome the first limitation, Park *et al.* [18] select K by checking the consistency of the reconstructed trajectory in an N-cross validation scheme. Alternatively, Valmadre *et al.* [5] propose 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.* [17] propose to incorporate the 3D structures of a number of key frames to enhance the reconstructability. However, obtaining those key-frame 3D structures requires manual interaction. The methods in [4], [5], [17], [18] 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 [17], [18].

Zheng *et al.* [6] 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 suffers from low scalability and efficiency due to the NP-hard GMST problem.

2.2 Non-rigid SfM

One class of related works solve the non-rigid structure from motion (NRSFM) problem, which targets simultaneous recovery of camera motion and 3D structure using an image sequence. The work by Bregler *et al.* [19] tackles the NRSFM problem through matrix factorization, with the assumption that deforming non-rigid objects can be represented by a linear combination of low-order shape bases. It is later shown by Xiao *et al.* [20] that utilizing only orthogonality constraints on the camera rotation is not sufficient, and a basis prior is required to uniquely determine the shape bases. Not until very recently, Dai *et al.* [9] propose a new prior-free method that minimizes the nuclear norm of the shape matrix. Based on this work, Zhu *et al.* [10] and Kong *et al.* [11] further improve the reconstruction accuracy by exploring a union of subspaces. Their methods relate to our approach in terms of leveraging subspace clustering (through low rank representation) and dictionary learning.

At first glance, it seems that the shape-based approaches can be applied to our problem without much modification. However, these approaches assume orthographic or weak perspective camera models, and it has been shown empirically that the extension of these methods to projective camera model is not straightforward [4]. There are works for projective non-rigid shape and motion recovery based on tensor estimation [8], [21], but this challenging problem is still under on-going research.

The methods for trajectory triangulation and NRSFM typically start from a set of 2D correspondences across frames. The 2D correspondences could be attained by optical flow [22] or feature based matching [23]. If the target object is people, the joint

detector could also be applied [24], [25]. In our paper, we assume 2D correspondences across frames are already available.

2.3 Sequencing and synchronization

Sequencing information is important in trajectory triangulation. Recently, Basha *et al.* [26], [27] 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 [26] relies on two images taken from roughly the same location in order to eliminate the uncertainty in the sequencing. Basha *et al.* [27] later introduce a solution that leverages the known temporal order of the images within each camera. Both of these methods assume dynamic objects move closely to a straight line within a short time period, but the assumptions may fail in practice. Video synchronization has attracted much attention in the computer vision community [28], [29], [30], but those methods have various constraints such as camera motion, availability of sound, and number of videos. In this paper, our approach aims at dynamic 3D reconstruction without sequencing across videos.

3 Problem and Notations

We now describe the notations of our problem. Let $\mathcal{I}$ denote an aggregated set of images obtained 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 assume an *a priori* camera registration through structure-from-motion analysis of static background structures within the environment [31]. 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 first assume each image I_f captures a common set of P 3D points $\{\mathbf{X}_{(p,f)} \mid p = 1, \ldots, P\}$, and the 2D measure of each point is denoted as $\mathbf{x}_{(p,f)}$. We also assume the correspondences of image measures $\mathbf{x}_{(p,f)}$ across images are available. Then for each measure $\mathbf{x}_{(p,f)}$, we can compute a viewing ray with direction by $\mathbf{r}_{(p,f)} = \mathbf{R}_f^{\mathrm{T}}\mathbf{K}_f^{-1}[\mathbf{x}_{(p,f)}^{\mathrm{T}} \;\; 1]^{\mathrm{T}}$, and followed with a normalization to a unit vector.

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

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

where $d_{(p,f)}$ 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.

Then by defining $\mathbb{C}$, $\mathbb{r}$, and $\mathbb{d}$ as follows,

$$\mathbb{C} = \begin{bmatrix} \mathbf{C}_1 & \cdots & \mathbf{C}_F \end{bmatrix}, \tag{3}$$

$$\mathbb{r} = \begin{bmatrix} \mathbf{r}_{(1,1)} & \cdots & \mathbf{r}_{(1,F)} \\ \vdots & \ddots & \vdots \\ \mathbf{r}_{(P,1)} & \cdots & \mathbf{r}_{(P,F)} \end{bmatrix}, \tag{4}$$

$$\mathbb{d} = \begin{bmatrix} d_{(1,1)} & \cdots & d_{(1,F)} \\ \vdots & \ddots & \vdots \\ d_{(P,1)} & \cdots & d_{(P,F)} \end{bmatrix}, \tag{5}$$

Eq. (1) for all the points can be rewritten in matrix form as

$$\mathbb{X} = \mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r}, \tag{6}$$

where $\mathbf{1}_{P\text{x}1}$ is a P-by-1 matrix with values equal to 1, $\otimes$ is the Kronecker product, and $\odot$ is the component-wise matrix product.

Our task is to recover $\mathbb{X}$ from the 2D measures without image sequencing information across the videos.

4 Principle

The key observation driving our approach is that dynamic shape exhibits temporal coherence. In this section, we demonstrate how this principle can be leveraged to recover local temporal ordering with known shapes. Our proposed method will extend these ideas to situations with unknown structures.

For our method, we assume a smooth 3D motion under the sampling provided by the videos. Hence, we can approximate the 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. That is, we have

$$\mathbf{S}_f \approx w \cdot \mathbf{S}_{prev} + (1 - w) \cdot \mathbf{S}_{next}, \tag{7}$$

with $0 \leq w \leq 1$. 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 reconstructability constraints make single-camera structure estimation ill-posed (see Sec. 8.2 for details). Hence, *we rely on inter-sequence temporal ordering information to solve the dynamic structure estimation problem*. The absence of a global temporal ordering requires us to search for temporal adjacency relations across the different video streams having 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

$$\mathbf{S}_f = \mathbb{X}\mathbf{W}_f, \tag{8}$$

where $\mathbf{W}_f = \left(w_{(1,f)}, \ldots, w_{(f-1,f)}, 0, w_{(f+1,f)}, \ldots, w_{(F,f)}\right)^{\mathrm{T}}$ is a vector of length F representing the coefficients for the linear combination. Note that the f-th element in $\mathbf{W}_f$ 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 a few shapes in the close temporal neighborhood of $\mathbf{S}_f$ are likely to provide a good approximation, we expect the vector $\mathbf{W}_f$ to be sparse. Accordingly, we propose to find the local temporal neighborhood of a shape $\mathbf{S}_f$ through a compressive sensing formulation leveraging the l_1 norm:

$$\underset{\mathbf{W}_f}{\text{minimize}} \; ||\mathbf{S}_f - \mathbb{X}\mathbf{W}_f||_2^2 + \lambda||\mathbf{W}_f||_1, \tag{9}$$

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 $\mathbf{W}_f$ [32]. Moreover, we incorporate the desired properties of our linear combination framework (Eq. (7)) and reformulate Eq. (9) as

$$\begin{aligned} \underset{\mathbf{W}_f}{\text{minimize}} \quad & ||\mathbf{S}_f - \mathbb{X}\mathbf{W}_f||_2^2 \\ \text{subject to} \quad & \mathbf{W}_f \cdot \mathbf{1}_{F\times 1} = 1 \\ & \mathbf{W}_f \geq 0. \end{aligned} \tag{10}$$

The affine constraints of Eq. (10) constrain the variable $\mathbf{W}_f$ to reside in the simplex Δ_f defined as

$$\Delta_f \triangleq \{\mathbf{W}_f \in \mathbb{R}^F \text{ s.t. } \mathbf{W}_f \geq 0, w_{(f,f)} = 0 \text{ and } \sum_{j=1}^{F} w_{(j,f)} = 1\} \tag{11}$$

Despite the lack of an explicit l_1 norm regularization term in Eq. (10), as a variant of compressive sensing, the formulation still keeps the sparsity-inducing effect [32], [33]. This is true for the present problem, since we know a shape can be well represented by temporally close shapes. A similar formulation has been used in modeling archetypal analysis for representation learning [33]. There, the authors also provide a new efficient solver for this kind of problem.

Finally, we generalize our formulation from Eq. (10) to include all available structure estimates $\mathbf{S}_f$, with $f = 1, \ldots, F$, into the following equation

$$\begin{aligned} \underset{\mathbb{W}}{\text{minimize}} \quad & ||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\text{F}}^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F, \end{aligned} \tag{12}$$

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

As an illustration of the validity of our compressed sensing formulation, Fig. 3 shows the output of Eq. (12) on a real motion capture dataset given known 3D points $\mathbb{X}$. Although image sequencing is assumed unknown, we show results in temporal order for visualization purposes. The coefficients in $\mathbb{W}$ approximate a matrix having non-vanishing values only on the locations directly above and below the main diagonal. This indicates that the 3D points $\mathbf{S}_f$ are a linear combination of $\mathbf{S}_{f-1}$ and $\mathbf{S}_{f+1}$.

Minimizing Eq. (10) is equivalent to finding the 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 local temporal information is recoverable from the non-vanishing values in $\mathbb{X}$. However, if object motion is repetitive or if the object is static for a period of time, there is no guarantee that the most related shapes are the temporally closest ones. Even though this is true, the analysis in Sec. 8.3 shows that this does not cause any problem for our method in regard to 3D reconstruction.

To validate our prior of sparse representation for real motion, we quantitatively evaluate the estimated coefficients $\mathbb{W}$ by minimizing Eq. (12) on all 130 real motion capture datasets presented in [34]. 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 obtain are 0.9972 and 0.9994, supporting the validity of our prior.

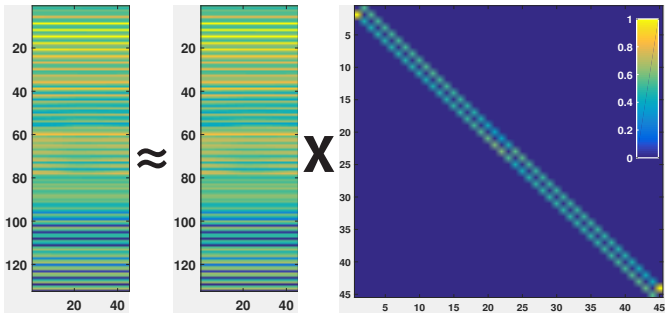


Fig. 3: We illustrate the output of Eq. (12) on a real motion capture dataset. For easy visualization, the shortest motion capture dataset (45 frames) presented in [34] is used. Each element/column in $\mathbb{X}$ corresponds to ground truth 3D structure. The estimation of $\mathbb{W}$ through Eq. (12) approximates the correct ordering after enforcing all elements in the diagonal to be 0.

5 METHOD

We address the problem of estimating sparse dynamic 3D structure from a set of spatially registered video sequences with unknown temporal overlap. Sec. 4 presented a compressive sensing formulation leveraging the self-expressiveness of all the shapes in the context of known 3D geometry, but our goal is to estimate the unknown structure. To this end, we define our dictionary as the temporally varying 3D structure and pose the estimation of 3D structure as a dictionary learning problem. This is achieved through the optimization of a biconvex cost function that leverages the compressive sensing formulation described in Sec. 4 and, additionally, enforces both structural dependence coherence across video streams and 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 $\mathbb{W}$, we extend our formulation from Equation (12) to the following cost function:

$$\begin{aligned} \underset{\mathbb{X},\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\text{F}}^2 + \lambda_1\Psi_1(\mathbb{W}) + \lambda_2\Psi_2(\mathbb{X}) \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F, \end{aligned} \tag{13}$$

where $\Psi_1(\mathbb{W})$ and $\Psi_2(\mathbb{X})$ are two convex cost terms regulating the spatial relationships between 3D observations within and across video streams. We also add the normalization term FP to cancel the influence of number of frames and number of points per shape. Next, we describe each of the cost terms in detail.

5.2 Dictionary space reduction in self-representation

The first cost term in Eq. (13) serves to find shapes in the dictionary to sparsely represent each shape. The search space can be reduced if some elements of $\mathbb{W}$ are forced to be 0. As mentioned, the diagonal elements of $\mathbb{W}$ are forced to be 0, since a shape is not used to represent itself. Moreover, it is possible that if *a priori* knowledge of rough temporal information across video steams is available, we can also leverage that knowledge to reduce the search space.

In our solution, 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 Sec. 8.2

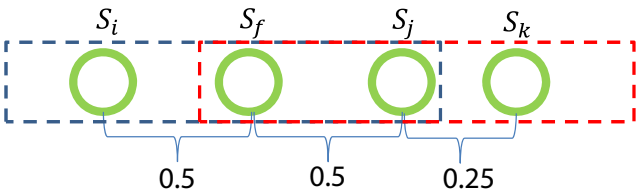


Fig. 4: Illustration of the triplets influencing the weights for $\mathbf{S}_f$ and $\mathbf{S}_j$ leading to an asymmetric $\mathbb{W}$. The values in the figure represent the distance between adjacent points.

shows such estimation is ill-posed. In our implementation, enforcing this constraint is achieved by not defining the corresponding variables in $\mathbb{W}$ during the optimization.

5.3 Coefficient relationships: $\Psi_1(\mathbb{W})$

As described in Sec. 4, a given structure $\mathbf{S}_f$ in frame f can be obtained from the linear combination of the 3D shapes captured in other frames. The coefficients or weights of the linear combination are given by the elements of the matrix $\mathbb{W}$. In particular, the element in the j-th row and f-th column of $\mathbb{W}$ is denoted as $w_{(j,f)}$, and it describes the relative contribution (weight) from $\mathbf{S}_j$ in estimating $\mathbf{S}_f$. Similarly, $w_{(f,j)}$ represents the contribution of $\mathbf{S}_f$ towards the 3D points in $\mathbf{S}_j$. Accordingly, a value of $w_{(f,j)} = 0$ indicates the absence of any contribution from $\mathbf{S}_f$ 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 correlated, 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 toward enforcing this property within the estimation of $\mathbb{W}$. We encode this relationship into our cost function as an additional term of the form

$$\Psi_1(\mathbb{W}) = \frac{1}{F}||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2. \tag{14}$$

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 $w_{(f,j)}$ and $w_{(j,f)}$ 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. In the toy 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.*, $w_{(i,f)} = w_{(j,f)} = \frac{1}{2}$. However, in order to determine the linear combination weights for specifying $\mathbf{S}_j$, we need to consider $\mathbf{S}_f$ and $\mathbf{S}_k$. Here, $\mathbf{S}_f$ is twice as far from $\mathbf{S}_j$ as $\mathbf{S}_k$, and thus $w_{(f,j)} = \frac{1}{3}$, which is lower than $w_{(j,f)}$. Accordingly, we do not expect a fully symmetric weight matrix $\mathbb{W}$.

Our formulation favors the assumption of local smooth motion (as in [5]), which mitigates the ambiguity raised from unary temporal observations in dynamic object reconstruction. Also, given our expectation of a sparse coefficient matrix $\mathbb{W}$, we can focus on finding congruence between the zero-value elements of the $\mathbb{W}$ and $\mathbb{W}^\top$, which $\Psi_1(\mathbb{W})$ effectively encodes.

5.4 Sequencing information: $\Psi_2(\mathbb{X})$

Under the assumption of sufficiently smooth 3D motion w.r.t. the frame-rate 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 $\mathbb{X}$

$$\Psi_2(\mathbb{X}) = \frac{1}{M}\sum_{n=1}^{N}\sum_{m=1}^{|\mathcal{V}_n|-1} ||\mathbf{S}_{(n,m)} - \mathbf{S}_{(n,m+1)}||_2^2, \tag{15}$$

where n is the video index, m is the image index within a video, $|\mathcal{V}_n|$ denotes the number of video frames within sequence n, and $M = \sum_{n=1}^{N}(|\mathcal{V}_n| - 1)$ is a normalization factor. 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, 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. (13) until convergence to obtain values for $\mathbb{X}$ and $\mathbb{W}$, and then taking those values as initialization, we further optimize Eq. (13) setting $\lambda_2 = 0$, effectively discarding $\Psi_2(\mathbb{X})$.

6 Parameterization of $\mathbb{X}$

Eq. (6) explicitly constrains the 3D structures $\mathbb{X}$ to lie on the viewing rays defined by the 2D measurements and camera poses. This corresponds to an implicit assumption of noise-free measurements. However, 2D feature measurements can be subject to localization inaccuracies or, in extreme cases, detection failure due to image capture aberrations (*e.g.*, motion blur or non-linear camera gain). Next, we discuss the parameterization of $\mathbb{X}$ given noisy and missing 2D observations.

6.1 Noisy observations

The parameterization using Eq. (6) enforces the hard constraint that 3D points lie on the viewing rays. Given that this may not be appropriate under the circumstance of noisy measurements, we propose a soft constraint by adding a regularization term into Eq. (13). Defining the objective function in Eq. (13) as $\Phi(\mathbb{X}, \mathbb{W})$, we propose a revised version as

$$\begin{aligned} \underset{\mathbb{X},\mathbb{W},\mathbb{d}}{\text{minimize}} \quad & \Phi(\mathbb{X}, \mathbb{W}) + \lambda_3||\mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r} - \mathbb{X}||_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F. \end{aligned} \tag{16}$$

The formulation converts the hard constraint of Eq. (6) as a soft constraint, by adding a penalization whenever the 3D points deviate from the viewing ray. The value of λ_3 controls how much a point can deviate away from the viewing ray, and it depends on the noise level of the 2D observations. Moreover, in Eq. (16), $\mathbb{d}$ is an auxiliary variable solely depending on $\mathbb{X}$. More details about the optimization of Eq. (16) are presented in Sec. 7.1.

6.2 Missing data

Each 3D point, given its accurate 2D measurement, lies on the corresponding viewing ray. Hence, the 3D point has one degree of freedom – depth along the ray. However, in the absence of 2D observations, which can happen in the case of occlusion, the 3D points are no longer constrained by the viewing ray and thus have three degrees of freedom.

In our method, the 3D points with missing 2D observations are interpolated by the estimated linear coefficients $\mathbb{W}$. Therefore, this scheme is likely to produce larger errors if a dynamic 3D point is not observed by multiple consecutive frames across time. In our experiments, we test the accuracy of our algorithm under different missing-data rates.

7 Optimization

The biconvex function in Eq. (13) is non-convex, but it is convex if one set of the variables $\mathbb{X}$ or $\mathbb{W}$ is fixed. The optimization scheme employed for Eq. (13) alternates the optimizations over $\mathbb{X}$ and $\mathbb{W}$. We preferred this approach due to its relative simplicity over elaborate dictionary update schemes such as K-SVD [12]. Nevertheless, since the alternating optimization steps need to be performed until convergence, each step must be reasonably fast. Although optimizing over $\mathbb{X}$ is easy, optimizing over $\mathbb{W}$ is relatively more difficult due to the simplicial constraint. We find that optimizing over $\mathbb{W}$ with a general solver, such as CVX [35], is too slow even for a moderate number of frames F. To solve the problem with speed and scalability, we propose a new solver based on alternating direction method of multipliers (ADMM) [36].

7.1 Optimize over $\mathbb{X}$

If $\mathbb{W}$ in Eq. (13) is fixed, the optimization over $\mathbb{X}$ is straightforward, as the problem is quadratic programming without any constraint, regardless of the difficulties discussed in Sec. 6.

1) If the data are noise-free, we can substitute Eq. (6) into Eq. (13), and obtain a quadratic programming problem without any constraint on the unknown variable $\mathbb{d}$.
2) In the case of noisy measurements, $\mathbb{d}$ are dependent on $\mathbb{X}$. More specifically, $d_{(p,f)}$ is given by
$$d_{(p,f)} = (\mathbf{X}_{(p,f)} - \mathbf{C}_f)^{\mathrm{T}}\mathbf{r}_{(p,f)}, \tag{17}$$
i.e. the projection of $\mathbf{X}_{(p,f)} - \mathbf{C}_f$ onto the viewing ray. Then, after replacing $\mathbb{d}$ with $\mathbb{X}$, we obtain a quadratic programming problem over unknown $\mathbb{X}$.
3) For the case of missing observations, the corresponding 3D points are unknown variables with three degrees of freedom. Therefore, under a given miss rate, the problem is quadratic over some unknown variables both in $\mathbb{d}$ and in $\mathbb{X}$.

For the quadratic programming without constraints, the solution can be found at the zero value of the derivative of the cost function over the unknown variables.

7.2 Optimize over $\mathbb{W}$

The optimization over $\mathbb{W}$ is more complex mainly due to the simplex constraints. By fixing the variable $\mathbb{X}$ in Eq. (13), the cost function becomes,

$$\begin{aligned} \underset{\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \frac{\lambda_1}{F}||\mathbb{W} - \mathbb{W}^\top||_2^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F. \end{aligned} \tag{18}$$

Notice that if the term $||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2$ vanishes, the cost function is the same to Eq. (12), which can be decomposed into Eq. (10), and optimized over $\mathbf{W}_f$ for each $f = 1, \ldots, F$ independently. Advantageously, the number of variables for each subproblem is much smaller compared to the total number of variables in $\mathbb{W}$, and it can be parallelized on the level of subproblems. Moreover, Chen *et al.* [33] propose a fast solver to the optimization problem in Eq. (10) based on an active-set algorithm that can benefit from the solution sparsity. However, the cost term $||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2$ prevents the decomposition.

In this paper, we propose an ADMM algorithm that enables the decomposition. By introducing a new auxiliary variable $\mathbb{Z}$, Eq. (18) can be rewritten as

$$\begin{aligned} \underset{\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \frac{\lambda_1}{F}||\mathbb{Z} - \mathbb{Z}^\top||_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F \\ & \mathbb{W} = \mathbb{Z}. \end{aligned} \tag{19}$$

Though this change may seem trivial, the objective function is now separated in $\mathbb{W}$ and $\mathbb{Z}$. The ADMM technique allows this problem to be solved approximately by first solving for $\mathbb{W}$ with $\mathbb{Z}$ fixed, then solving for $\mathbb{Z}$ with $\mathbb{W}$ fixed, and next proceeding to update a dual variable $\mathbb{Y}$ (introduced below). This three-step process is repeated until convergence. Next, we describe each step of our ADMM-based algorithm.

In step 1, $\mathbb{W}$ is updated by

$$\begin{aligned} \mathbb{W}^{k+1} = \underset{\mathbf{W}_f \in \Delta_f, \text{ for} 1 \le f \le F}{\operatorname{argmin}} & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 \\ & + \mathrm{vec}(\mathbb{Y}^k)^\top \mathrm{vec}(\mathbb{W}) + \frac{\rho}{2}||\mathbb{W} - \mathbb{Z}^k||_{\mathrm{F}}^2, \end{aligned} \tag{20}$$

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 $\mathbb{W}$. The optimization can be decomposed into optimizing over $\mathbf{W}_f$ independently and in parallel, and we employ the fast solver proposed in [33].

In step 2, we update the auxiliary variable $\mathbb{Z}$ according to

$$\begin{aligned} \mathbb{Z}^{k+1} = \underset{\mathbb{Z}}{\operatorname{argmin}} & \frac{\lambda_1}{F}||\mathbb{Z} - \mathbb{Z}^\top||_{\mathrm{F}}^2 - \mathrm{vec}(\mathbb{Y}^k)^\top \mathrm{vec}(\mathbb{Z}) \\ & + \frac{\rho}{2}||\mathbb{W}^{k+1} - \mathbb{Z}||_{\mathrm{F}}^2. \end{aligned} \tag{21}$$

This is a quadratic programming problem in the unknown variable $\mathbb{Z}$ without constraint and can be easily solved by setting the derivative of Eq. (21) with respect to $\mathbb{Z}$ equal to 0.

In step 3, the dual variables $\mathbb{Y}$ are updated directly by

$$\mathbb{Y}^{k+1} = \mathbb{Y}^k + \rho(\mathbb{W}^{k+1} - \mathbb{Z}^{k+1}). \tag{22}$$

The three Eqs. (20), (21), and (22) iterate until the stop criterion is met. We use the stop criterion described in [36].

7.3 Initialization of the Optimization

Given the non-convexity of our original cost function (Eq. (13)), the accuracy of our estimates is sensitive to the initialization values used by our iterative optimization. Hence, we design 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 2D 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 $\mathbf{X}_{(p,f)}$. 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 order to avoid a trivial solution arising from the small-baseline typically associated with

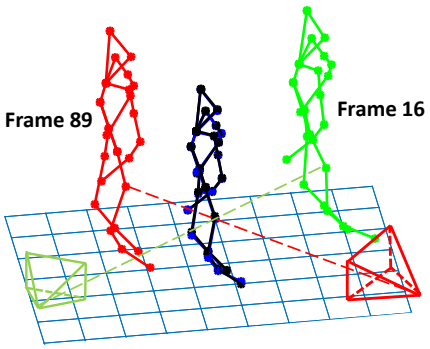


Fig. 5: Example of incorrect initialization. The dataset 'hopBothLegs3hops' in [34] has the motion of hopping forward three times. The black and blue shapes (almost overlapped) are the incorrect initialization of the real shapes (shown in green and red) of frames 16 and 89 due to the accidental ray intersections. This typically happens in the case of periodic motion such as walking or jogging. In the figure, only one set of nearly intersecting rays is plotted.

consecutive frames of a single video, we restrict our search to ray bundles captured from distinct video sequences.

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

$$\{\mathbf{d}_f^*, \mathbf{d}_j^*\} = \underset{\mathbf{d}_f, \mathbf{d}_j}{\operatorname{argmin}} ||\mathbf{S}_f - \mathbf{S}_j||_2^2. \tag{23}$$

This is a quadratic cost function with a closed-form solution.

We then build a symmetric distance matrix $\mathbf{D}$ with element $D_{(f,j)}$ equal to the minimum cost of Eq. (23). If the frames f and j are from the same video, $D_{(f,j)}$ is set to infinity. Next, we identify many pseudo-intersection points with negative depth (*i.e.* divergent pairs of viewing rays), and set the corresponding element in $\mathbf{D}$ 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 $\mathbf{S}_f$.

The above initialization is done regardless of available measurements, since we only look for an approximate initialization for the solver. In the case of missing data, the corresponding 3D points in the shape are simply ignored when minimizing Eq. (23).

The output of the initialization is typically close to the ground truth, but may fail occasionally, as is shown in Fig. 5. This kind of wrong initialization may lead to wrong estimation of the two shapes if the smoothness term $\Phi_2(\mathbb{X})$ in Eq. (13) is not present, because these two shapes can well represent each other. Our cost term $\Phi_2(\mathbb{X})$ helps to pull the occasional incorrect shapes out of local minima.

8 Analysis and Discussion

This section provides key insights to our algorithm for dynamic object reconstruction without sequencing. The following statements will be illustrated in detail.

1) Interleaved 2D measures across video streams yields favorable viewing ray geometry for 3D shape estimation.
2) High-frequency 2D observations and smooth object motion jointly validate our self-expressive structure prior for accurate shape estimation.
3) No dependence on the availablity of sequencing information as opposed to existing approaches [4], [5].

Next, we first describe the formulation of reconstruction errors by our method, based on which the above statements are illustrated at length in the subsequent three subsections.

8.1 Representation of Reconstruction Errors

Our solution computes 3D structure by minimizing the non-convex function Eq. (13). Since direct analysis of the non-convex function is difficult, we assume $\mathbb{W}$ is already known and fixed. Without loss of generality, we also assume the 2D observations are noise-free.

Given that in our method λ_2 is set to 0 in the end, and $\mathbb{W}$ is known and fixed, Eq. (13) is equivalent to

$$\underset{\mathbb{X}}{\text{minimize}} \quad ||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\text{F}}^2. \tag{24}$$

From Eq. (24), it can be seen when $\mathbb{W}$ is fixed, all points in a shape are computed independently, and computing one 3D point per shape versus multiple points per shape basically follows the same routine. Therefore, for the sake of more concise presentation, the analysis in this section assumes only one point per shape, and the point index p for the shape is omitted.

To analyze the reconstruction error, we assume that the ground truth of the 3D points is already known, and then analyze how much the computed structure deviates away from the ground truth, which is deemed as reconstruction error. We denote the ground truth 3D point as $\mathbb{X}^* = [\mathbf{X}_1^*, \cdots, \mathbf{X}_f^*, \cdots, \mathbf{X}_F^*]$. Then, any point $\mathbf{X}_f$ on the viewing ray that passes through $\mathbf{X}_f^*$ can be parameterized as

$$\mathbf{X}_f = \mathbf{X}_f^* + l_f \mathbf{r}_f, \tag{25}$$

where the unknown l_f is the signed distance from the ground truth along the viewing ray.

When minimizing Eq. (24), using either Eq. (25) or Eq. (1) to represent $\mathbf{X}_f$ in practice generates different values of d_f and l_f, but the estimated 3D points are actually identical. Therefore, $|l_f|$ represents the Euclidean error of our method.

Eq. (24) is a quadratic objective function without any constraint and has a closed-form solution. We use Eq. (25) to represent the 3D point, and by setting the derivative of Eq. (24) over variables $\mathbf{l} = [l_1, \dots, l_f, \cdots, l_F]$ to 0, we obtain a linear equation system denoted as

$$\mathbf{A}\mathbf{l} = \mathbf{b}, \tag{26}$$

where $\mathbf{A}$ is an $F \times F$ matrix with the f-th row given by

$$\mathbf{A}_{:f} = (\mathbf{I} - \mathbb{W})_{:f}(\mathbf{I} - \mathbb{W})^{\text{T}}\text{diag}([\mathbf{r}_1^T\mathbf{r}_f, \cdots, \mathbf{r}_F^T\mathbf{r}_f]), \tag{27}$$

and $\mathbf{b}$ is an $F \times 1$ vector with the f-th element given by

$$\mathbf{b}_f = \mathbf{r}_f^T\mathbb{X}^*(\mathbf{I} - \mathbb{W})(I - \mathbb{W})_{:f}^{\text{T}}. \tag{28}$$

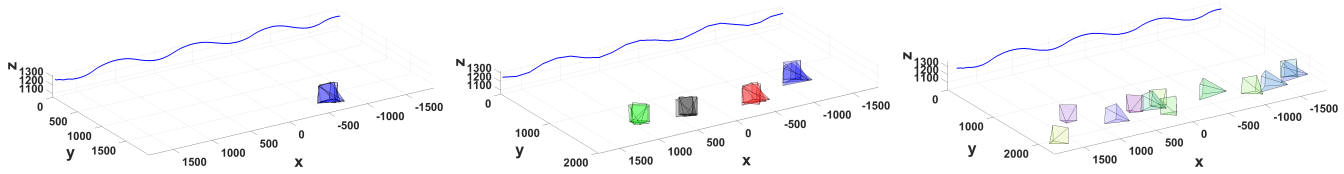
In Eqs. (27) and (28), the subscript $_{:f}$ denotes the f-th row of a matrix, and $\mathbf{I}$ is an identity matrix. Then the solution for $\mathbf{l}$ is

$$\mathbf{l} = \mathbf{A}^{-1}\mathbf{b}. \tag{29}$$

As mentioned, $\mathbf{l}$ is the reconstruction error, which is bounded by

$$||\mathbf{l}||_2 = ||\mathbf{A}^{-1}\mathbf{b}||_2 \leq ||\mathbf{A}^{-1}||_2||\mathbf{b}||_2. \tag{30}$$

A large upbound in Eq. (30) means unstable reconstruction results and typically larger errors. In this paper, we use the term reconstructability (first defined in [4]) as a criterion to characterize the reconstruction accuracy of our algorithm. In order to achieve high reconstructability, $||\mathbf{A}^{-1}||_2$ and $||\mathbf{b}||_2$ should be small. Next, we discuss $||\mathbf{A}^{-1}||_2$ and $||\mathbf{b}||_2$ in detail.



(a) System condition: 1.48e+04 (b) System condition: 22.3 (c) System condition: 29.0

Fig. 6: Simulated camera setups. The blue curve is a trajectory of a 3D point obtained from motion capture data. Figs. 6a and 6b depict the camera setups of one and four slow-moving handheld cameras. Fig. 6c depicts a scenario where each random camera only captures one image. Fig. 6b and Fig. 6c show the camera setups used in our method and [6], respectively. Coordinates are in millimeters (mm).

8.2 System condition

Based on the definition of the matrix Euclidean norm, we have

$$||\mathbf{A}^{-1}||_2 = 1/\sigma_{\min}, \tag{31}$$

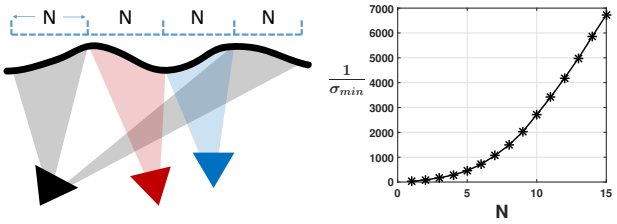
where $\sigma_{\min}$ is the smallest singular value of matrix $\mathbf{A}$. With fixed $\mathbb{W}$, we observe from Eq. (27) that $\mathbf{A}$ solely relies on the viewing ray directions and does not depend on the exact positions of the 3D points $\mathbb{X}^*$ along the viewing rays. Since $\sigma_{\min}$ is closely related to reconstruction errors and is determined by the camera system setup, we call it system condition. Note the system condition introduced here is in essence very similar to the system condition number described in the works [5], [6].

Since direct analysis of the system condition given viewing ray directions $\{\mathbf{r}_1, \ldots, \mathbf{r}_F\}$ based on Eq. (27) is difficult, we next use empirical simulation to demonstrate the system condition under different camera setups. In the experiments, we synthesize 2D input features from real motion capture datasets that sample the 3D structure of real dynamic objects at 40 Hz. Figs. 6a and 6b simulate setups of one handheld camera and multiple handheld cameras that record videos of a person walking. To mimic small random motion in each handheld camera, the camera centers at different time instances are perturbed by Gaussian noise with standard deviation of 10 mm around a fixed center. We also test the case of completely random cameras (Fig. 6c), with each taking one photo. The 3D structure at each time instance is projected to one of the virtual cameras to generate a set of 2D observations. For the scenario in Fig. 6b, we ensure no two shapes at consecutive time instances are projected into the same video stream. To compute $\sigma_{\min}$ in the simulation, we use the best possible estimate of $\mathbb{W}$ (denoted as $\mathbb{W}^*$), which is defined as the output of Eq. (13) given ground truth structure. We next show even with $\mathbb{W}^*$, accurate 3D reconstruction is not guaranteed.

We estimate the system condition using Eq. (31) on 500 trials with random cameras. The average system conditions in Fig. 6 show the setup with one handheld camera has very low reconstructability. Note that even though the system conditions of the camera setups in Figs. 6b and 6c are favorable, in practice the important sequencing information (see Sec. 8.4) across different cameras for these two cases is not readily available.

To illustrate the importance of cross-sequence 2D observations for our structure estimation process (statement 1), we evaluate system condition as a function of increased temporal gaps between cross-sequence samples. As shown in Fig. 7a, the dynamic object is observed by one camera for N frames, and then observed by another camera for N frames. We show empirically that as N increases, the system condition increases monotonically, which indicates more unstable reconstruction and typically larger errors (see experiments in Sec. 9.1.3), even under the assumption that $\mathbb{W}$ can be correctly estimated. This also illustrates that temporally consecutive shapes observed by the same video stream should not be used to represent each other, as is done in Sec. 5.2.

In fact, we observe that the reconstructability is closely related to the camera motion and the object motion. Specifically, if shape $\mathbf{S}_j$ is the most related shape to $\mathbf{S}_f$, as indicated by $\mathbb{W}$, the relative directions of viewing rays $\mathbf{r}_f$ and $\mathbf{r}_j$ (note we only have one point per shape in this analysis), determine the reconstructability. If the directions of $\mathbf{r}_f$ and $\mathbf{r}_j$ converge, *i.e.* the camera motion is relatively larger than the object motion, the reconstructability is higher. In the case of one handheld camera, the camera motion can be much smaller than the dynamic objects, and the viewing rays diverge, yielding low reconstructability. In contrast, if $\mathbf{r}_j$ and $\mathbf{r}_f$ are associated with different video cameras, the distance between the camera centers is much larger than the motion of the object. Hence the reconstructability is high. This observation is analogous to the classic triangulation of static scenes, where small baselines produce inaccurate reconstruction. Note the same conclusion was also made by Park *et al.* [18], though their reconstruction algorithm is different from ours.



(a) Observation without overlap (b) System condition

Fig. 7: The reconstructability of the system is lower if the period of single-camera capture is longer.

8.3 Shape approximation residual

While $\mathbf{A}$ depends on the viewing ray directions, which are available before reconstruction, $\mathbf{b}$ relies on the actual unknown positions of the ground truth structure $\mathbb{X}^*$ (Eq. (28)). To achieve accurate reconstruction, $||\mathbf{b}||_2$ should be close to 0.

Since in Eq. (28), $(I - \mathbb{W})^{\mathrm{T}}_{:f}$ is sparse, $\mathbf{b}_f$ can be considered as a linear combination of a few columns of matrix $\mathbb{X}^*(I - \mathbb{W})$ multiplied using dot product with the unit vector $\mathbf{r}_f$. Therefore, the value of $\mathbf{b}_f$ and hence the magnitude of $||\mathbf{b}||_2$, mainly rely on $||\mathbb{X}^*(I - \mathbb{W})||_{\mathrm{F}}$. Accordingly, we define the residual per point as

$$res = \frac{1}{PF}||\mathbb{X}^*(I - \mathbb{W})||_{\mathrm{F}}. \tag{32}$$

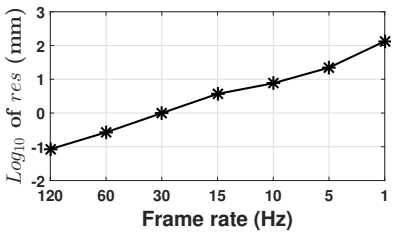


Fig. 8: Average residuals res at different camera frame rates. Results are attained from 130 motion capture datasets in [34].

The residual res is small if all the shapes can be well represented by other shapes. It relies on speed of object motion and the capturing frame rate. Using the best possible estimate of coefficients (*i.e.* $\mathbb{W}^*$), we test the residual res given motion capture data sampled at different frame rates. Fig. 8 shows res becomes larger as the frame rate goes down. This fits the intuition that shapes that are tempo-spatially farther away are less correlated. This also implies that our method cannot achieve accurate reconstruction from discrete images with large temporal discrepancy.

In practice, the estimated $\mathbb{W}$ might be different from $\mathbb{W}^*$. It is possible the estimated $\mathbb{W}$ associates to the current shape a set of two or more spatially proximal shapes, instead of the two temporally immediate neighbors, while still achieves a small residual value per Eq. (32). This could happen in case of linear object motion, objects being static, or noisy 2D measurements. Moreover, in the case of repetitive motion, such as a person waving his hand, it is possible the estimated $\mathbb{W}$ picks temporally farther away but highly correlated shapes and the residual is small. This means our algorithm could possibly produce accurate 3D reconstruction but poor local temporal information.

8.4 Importance of image sequencing

The temporal order of images, *i.e.* image sequencing, plays an important role in dynamic object reconstruction [4], [5]. The work by Valmadre *et al.* [5] generalizes the method of [4] in a new framework based on high-pass filters. Here, we briefly describe the method in [5] and its relation to our method, from which it can be revealed why their methods [4], [5] require sequencing information as opposed to ours.

Assuming the object moves smoothly in the space, Valmadre *et al.* [5] triangulate the 3D trajectory of an 3D point by minimizing its response to a set of high-pass filters. Given a predefined high pass filter $\mathbf{g} = [g_M, \dots, g_1]$, the trajectory is estimated by

$$\underset{\mathbb{X}}{\text{minimize}} \; ||\mathbb{X}\mathbf{G}||_{\text{F}}^2, \tag{33}$$

where $\mathbf{G}$ is defined as

$$\mathbf{G} = \begin{bmatrix} g_M & & \\ \vdots & \ddots & \\ g_1 & \ddots & g_M \\ & \ddots & \ddots \\ & & g_1 \end{bmatrix}. \tag{34}$$

Each column of $\mathbf{G}$ is a high-pass filter for the local region of a smooth trajectory. From the method model, it requires all the shapes (columns of $\mathbb{X}$) and hence the 2D meansurements to be ordered temporally.

Comparing Eq. (33) with Eq. (24), we can see the two equations are the same if $\mathbf{G}$ equals $\mathbf{I} - \mathbb{W}$. In effect, the method in [5] can be regarded as our method with a predefined $\mathbb{W}$. For instance, if the high pass filter is set to $\mathbf{g} = [1, -1]$, it is equivalent (ignoring the difference at boundary) that $\mathbb{W}$ is set to

$$\mathbb{W} = \begin{bmatrix} 0 & & \\ 1 & 0 & \\ & 1 & \ddots \\ & & \ddots \end{bmatrix}. \tag{35}$$

Therefore, an alternative interpretation of their method [5] using the high-pass filter $\mathbf{g} = [1, -1]$ in terms of our theory is approximating the current shape using only the temporally closest shape. Another high-pass filter proposed in [5] is $\mathbf{g} = [-1, 2, -1]$, which in our case is equivalent to fixing the weights of two neighboring shapes to 0.5.

The importance of sequencing can also be revealed from analysis of residual defined by Eq. (32). For the method in [5] with predefined $\mathbf{G}$, the residual will be large if columns of $\mathbb{X}^*$ are randomly shuffled. In contrast, our method leverages compressive sensing to estimates $\mathbb{W}$ (instead of predefined), which automatically picks the most related shapes to produce small residuals.

9 Experiments

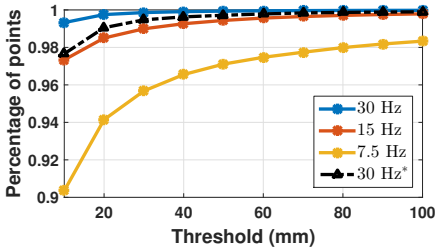
In our experiments, we evaluate our algorithm on both synthetic and real datasets. λ_1 and λ_2 in Eq. (13) are set empirically to 0.05 and 0.1 for all the experiments. To alleviate the influence of different camera system scales (*i.e.* differing the scale of $\mathbb{X}$), the average distance between camera centers is normalized to 1 before applying our method. The soft constraint parameterization is used only in the presence of noisy measurements.

9.1 Simulation

We use synthetic datasets to evaluate the accuracy and robustness of our proposal, and also compare against two state-of-the-art methods [5], [9]. To generate synthetic data, we use the real motion capture datasets from [34], and leverage them as ground truth structure for our estimation. The whole datasets contain 130 different real motions including hopping, jogging, cartwheel, punching, *etc*. Each motion capture dataset is comprised of the temporal sequences of a common set of 44 3D points in real scale, which corresponds within our framework to ground truth structure $\mathbb{X}_{GT}$. The frame rate of the motion datasets, *i.e.* the sampling rate of the real continuous motion, is 120 Hz. The length of each dataset ranges from 45 to 701 frames, and with an average of 273.

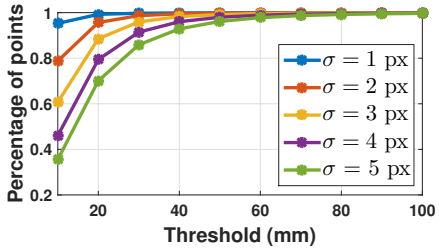
These 3D points are projected onto virtual cameras to generate input 2D measures into our methods. We select 4 virtual cameras with a resolution of 1M and focal length of 1000, and we position the static cameras around the centroid defined by $\mathbb{X}_{GT}$. The distance of the camera to the centroid is approximately twice the scale of $\mathbb{X}_{GT}$, and on average the distance is 2.7 meters. Considering the frame rate of the motion capture datasets is 120 Hz and there are 4 virtual cameras, the average frame rate for each camera is 30 Hz. Every temporal 3D capture is randomly assigned to each camera to build 4 disjoint image sequences. Unless otherwise mentioned, we enforce that no temporally consecutive captures are assigned to the same image sequence.

To evaluate our method, Euclidean errors between the ground truth and the estimated 3D points are computed. We define the accuracy by counting the percentage of points having errors less than thresholds of 10, 20, 30, 40, 50, and 100 mm.



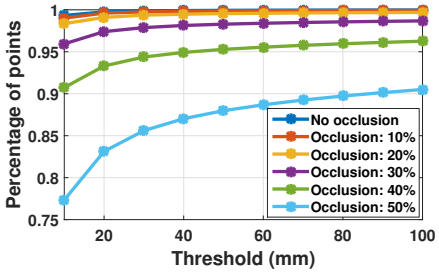
Frame rate \ Threshold	10	20	30	40	50	100
30	0.9933	0.9975	0.9986	0.9991	0.9994	0.9998
15	0.9734	0.9850	0.9899	0.9926	0.9944	0.9979
7.5	0.9036	0.9415	0.9568	0.9655	0.9711	0.9833
30* (unconstrained assignment)	0.9766	0.9905	0.9947	0.9963	0.9971	0.9990

Fig. 9: The reconstruction accuracy given different camera frame rates. We also test the case that the captures of object motion are randomly assigned to any of the image sequences without any constraint. 30 Hz* in the figure represents the unconstrained assignment.



Noise \ Threshold	10	20	30	40	50	100
$\mathcal{N}(0,1)$	0.9529	0.9925	0.9974	0.9987	0.9992	0.9998
$\mathcal{N}(0,2)$	0.7878	0.9568	0.9869	0.9949	0.9976	0.9997
$\mathcal{N}(0,3)$	0.6074	0.8855	0.9593	0.9828	0.9917	0.9991
$\mathcal{N}(0,4)$	0.4601	0.7941	0.9144	0.9602	0.9797	0.9980
$\mathcal{N}(0,5)$	0.3551	0.7008	0.8590	0.9287	0.9615	0.9966

Fig. 10: The reconstruction accuracy when the 2D observations are corrupted with Gaussian noise of different standard deviation (σ).



Miss rate \ Threshold	10	20	30	40	50	100
0%	0.9933	0.9975	0.9986	0.9991	0.9994	0.9998
10%	0.9901	0.9961	0.9975	0.9982	0.9986	0.9993
20%	0.9835	0.9910	0.9936	0.9948	0.9955	0.9968
30%	0.9594	0.9740	0.9788	0.9813	0.9829	0.9868
40%	0.9074	0.9331	0.9438	0.9493	0.9529	0.9626
50%	0.7734	0.8313	0.8560	0.8703	0.8798	0.9050

Fig. 11: The reconstruction accuracy under different percentages of occluded points.

9.1.1 *Accuracy*

Different frame rates. We first evaluate how the algorithm behaves under different capture frame rates. 2D measures without noise are used to evaluate the accuracy of our method. In addition to the original motion capture data at 120 Hz, we also downsample the data to 60 and 30 Hz, so that each camera has frame rate of 15 and 7.5 Hz on average. As shown in Fig. 9, the accuracy becomes worse as the frame rate gets slower. The main reason is that the self-representation residual is larger at lower frame rate. We notice that at a frame rate of 7.5 Hz, our method does not work well on the quick motions with large and nonlinear shape deformation, such as hopping or arms rotation. However, still more than 97% of 3D points have errors less than 5 cm, which is already very small considering the scale of a person and the distance range of the cameras.

Local temporal information. We also quantitatively evaluate the estimated $\mathbb{W}$. Using the same two measures described in Sec. 4, we get values of 0.9902 and 0.9923, compared to 0.9972 and 0.9994 if the 3D points are given. Therefore, our method very accurately recovers the local temporal information.

Unconstrained capture assignment. We test the case that each capture is randomly assigned to one of the four cameras so that temporally consecutive captures could have a chance to be assigned to the same camera, as is shown in Fig. 12. In this specific case, shapes $\mathbf{S}_1$ and $\mathbf{S}_5$ are used to represent $\mathbf{S}_2$, $\mathbf{S}_3$ and $\mathbf{S}_4$. Based on the theory in Sec. 8.3, using spatially further away shapes to represent the current shape has larger residual and hence larger reconstruction errors, as is validated in Fig. 9.

9.1.2 *Data robustness*

To evaluate the robustness of our method, we test it in the case of noisy measurements and missing data.

Noisy measurements. We add zero-mean Gaussian noise with different standard deviations to the 2D measurements. Considering that the focal length of the image is 1000 pixel, one pixel error corresponds to one millimeter if the object is one meter away. We apply the soft constraint formulation described in Sec. 6.1 and empirically set the parameter λ_3 to 100. As depicted in Fig. 10, the quality of reconstruction degrades as the noise level increases. As λ_3 increases, the soft constraint approximates the hard constraint. We evaluate the difference of the estimated results by the hard constraint formulation and the soft constraint formulation with different λ_3, and we show the median difference in Fig. 13. It is apparent that as λ_3 increases, the difference of the output between the two formulations becomes smaller.

We have tested the hard constraint formulation using noisy measurements, and the overall accuracy of the output is very similar. Though the soft constraint appears more robust in the presence of noise as it allows the points off the viewing ray, there is no guarantee or proof this constraint will achieve more accurate results, as it depends on the exact motion of the objects.

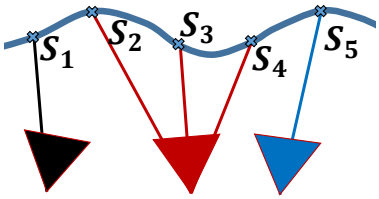


Fig. 12: Consecutive captures are assigned to the same red camera. For easy visualizations, only one point per shape is drawn.

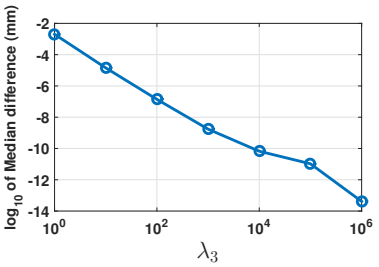


Fig. 13: The difference of the estimated results by the hard constraint formulation in Eq. (6) and the soft constraint formulation in Eq. (16) with different λ_3.

We observe that in the case of noisy measurements, the local temporal information cannot be well recovered. The temporally close shapes, in addition to the temporally immediate neighbors, could also contribute to the representation of the current shape. This is because under noisy measurements, the reconstructed shape at each time instance is less distinctive. Depending on the speed and types of the object motion, the time stamp of the contributing shapes could vary significantly. However, Fig. 10 still demonstrates reconstruction of hight accuracy, validating our analysis in Sec. 8.3.

Missing data. In our evaluation, we randomly set some 2D measures to be unavailable. Fig. 11 depicts the accuracy under different percentages of missing data. We observe that under 20% of occlusion, there is not much difference in reconstruction accuracy. Moreover, under a large amount of 40% occlusion, our method still produces accurate results, with 94.38% of points having errors less than 30 mm. Our method essentially linearly interpolates the 3D points along the trajectory using estimated $\mathbb{W}$. It can still produce 3D estimates in the presence of consecutive missing observations across time, but the accuracy in such scenarios depends on the object motion. Particularly, given large displacement of nonlinear motion, our method is likely to produce less accurate results.

9.1.3 Comparison to other methods

We compare our method with a NRSFM method [9] and A trajectory triangulation method [5]. Both of these methods are state-of-the-art for dynamic object reconstruction.

NRSFM method. It is tempting to use non-rigid structure from motion (NRSFM) methods to solve our problem, since our problem with known camera poses seems to be easier. However, most NRSFM methods work on an orthographic or weak perspective camera model, and it is unclear of their applicability under the perspective camera model. Park *et al.* [4] test the NRSFM methods [37], [38], [39] under a perspective camera model, but all of them fail to produce reasonably good results. In this paper, we test the state-of-the-art NRSFM method by Dai *et al.* [9].

The method by Dai *et al.* [9] recovers the shape matrix (corresponding to $\mathbb{X}$ in our problem description) by minimizing the matrix nuclear norm. Their method applies to an orthographic camera model, but can be easily adapted to a perspective model, as described below. Denoting

$$\mathbb{X}^{\#} = \begin{bmatrix} X_{(1,1)} & \dots & X_{(P,1)} & Y_{(1,1)} & \dots & Y_{(P,F)} & Z_{(1,1)} & \dots & Z_{(P,F)} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & & \\ X_{(1,F)} & \dots & X_{(P,F)} & Y_{(1,F)} & \dots & Y_{(P,F)} & Z_{(1,1)} & \dots & Z_{(P,F)} \end{bmatrix},$$

where $\mathbf{X}_{(p,f)} = (X_{(p,f)}, Y_{(p,f)}, Z_{(p,f)})^{\mathrm{T}}$. The shape of the object can be recovered through

$$\begin{aligned} \underset{\mathbb{X}^{\#},\mathbb{X}}{\text{minimize}} \quad & ||\mathbb{X}^{\#}||_* + \mu||\mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r} - \mathbb{X}||_{\mathrm{F}} \\ \text{subject to} \quad & \mathbb{X}^{\#} = \mathcal{L}(\mathbb{X}), \end{aligned}$$

where $||\cdot||_*$ is the matrix nuclear norm, μ is a positive weight, and $\mathcal{L}$ is a linear operator that reshapes $\mathbb{X}$ into $\mathbb{X}^{\#}$.

This formulation seems attractive at first glance due to its convexity, in contrast to our non-convex formulation. Moreover, their method is shape-based (instead of trajectory-based), and does not require temporal information. To test the NRSFM method, we use synthetic data without noise and the random camera configuration shown in Fig. 6c. Unfortunately, the results in Fig. 14b show that it completely fails, as opposed to our method shown in Fig. 14a.

Trajectory triangulation method. We also compare with the trajectory triangulation method by Valmadre *et al.* [5], as is described in Sec. 8.4. Since the required sequencing information is readily available within each video stream, our test uses the simulation of one handheld camera as shown in Fig. 6a. The camera centers are Gaussian with 20 mm standard deviation (σ_c) around a fixed point. Based on the theory in Sec. 8.2, the reconstructability increases with larger σ_c. Considering that the framerate of the motion capture dataset is 120 Hz, the camera motion with $\sigma_c = 20$mm is already very large compared to real handheld captures.

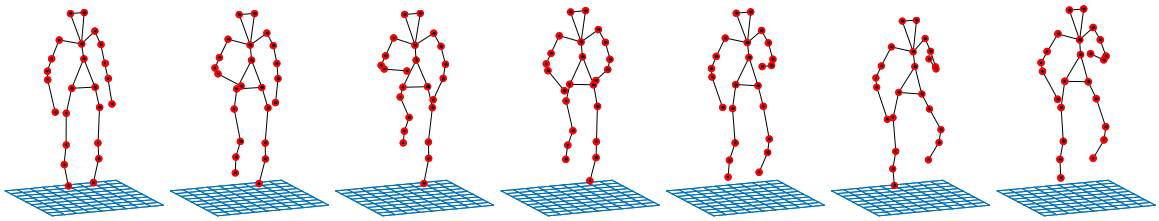
The method triangulates the trajectory of each dynamic point independently, and each trajectory has one system condition given the viewing ray directions. Since the motion of the person's head is relatively slower than that of his legs, the corresponding system condition is lower and the reconstructed points are more accurate, based on the theory in Sec. 8.2. Fig. 14c shows the large system condition ($1/\sigma_{\text{min}} = 2228$) in this camera setup leads to significant reconstruction errors.

9.2 Real datasets

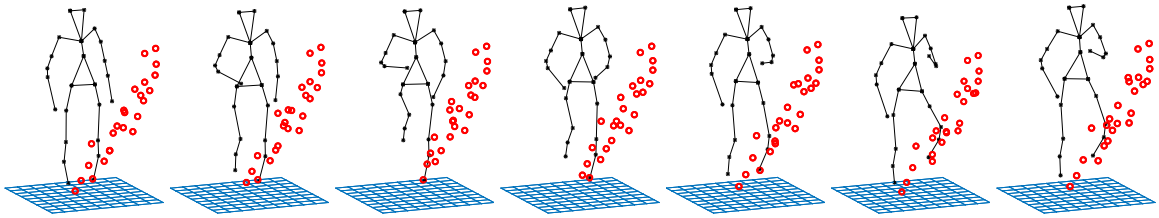
For experiments on real image capture, we use the Juggler and Rothman datasets from [40]. Given that the original datasets were synchronized, we sample the video frames to avoid concurrent captures (see Fig. 15). We do not use the datasets in [4], [26] because they only provide images with large temporal discrepancy, and therefore the shape residual is large (*i.e.* Eq. (7) does not hold). We also capture a new dataset of a person juggling using three iPhone6 and one iPhone5 without temporal synchronization.

We perform manual feature labeling on the input sequences and provide the obtained set of 2D measurements as input for our estimation process. For visualization purposes, Figs. 15 and 16 depict the estimated 3D geometry by connecting the estimated position of the detected joint elements through 3D line segments.

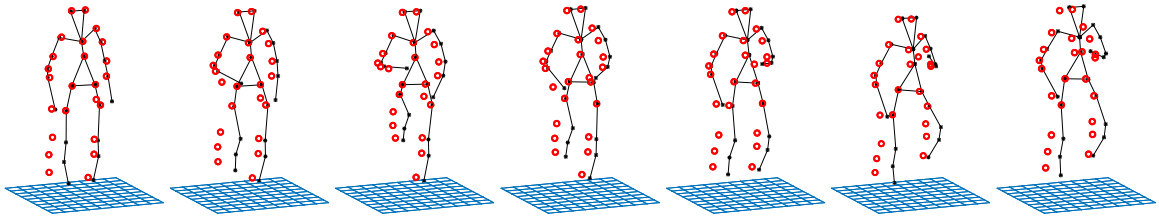
Our method also applies to multi-object reconstruction, as all the points on different objects together can be regarded as one shape, and the principle in Sec. 4 still holds. To demonstrate that, Fig. 16 reconstructs both 3D skeleton points and the juggler balls.



(a) Our method accurately reconstructs the 3D points ($1/\sigma_{\min} = 7.589$, $err = 0.0825$).



(b) The modified prior-free method [9] fails to produce reasonable results. ($err = 472.9033$)



(c) General trajectory prior method [5] produces large errors due to high system condition ($1/\sigma_{\min} = 2228$, $err = 76.9700$).

Fig. 14: Qualitative comparison of our method with [9] and [5] on the motion capture dataset 'jog on place' in [34]. The dataset has 214 frames, with 44 points per frame (only 24 are shown for visualization purposes). The black and red points are the ground truth and the estimated results, respectively. err is the average Euclidean error per point.

10 Conclusion and Contributions

This paper addresses the problem of dynamic 3D reconstruction using unsynchronized cross-video streams. The contributions of our framework encompass:

1) **Methodology Formulation**. We pose the problem in terms of a self-expressive dictionary learning framework leveraging a novel data-adaptive local 3D interpolation model.
2) **Implementation Mechanisms**. We define and solve a biconvex optimization problem and develop an efficient ADMM-based solver amenable for parallel implementation.
3) **Reconstructability Analysis**. We closely relate the reconstruction accuracy with both the system condition and the shape approximation residuals.

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

References

[1] E. Zheng, D. Ji, E. Dunn, and J. Frahm, "Sparse dynamic 3d reconstruction from unsynchronized videos," in *ICCV*, 2015.

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

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

[4] 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.

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

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

[7] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *IJCV*, 1992.

[8] R. Hartley and R. Vidal, "Perspective nonrigid shape and motion recovery," in *ECCV*, 2008.

[9] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *IJCV*, 2014.

[10] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey, "Complex non-rigid motion 3d reconstruction by union of subspaces," in *CVPR*, 2014.

[11] C. Kong and S. Lucey, "Prior-less compressible structure from motion," in *CVPR*, 2016.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, 2006.

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

[14] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009.

[15] V. Ramakrishna, T. Kanade, and Y. Sheikh, "Reconstructing 3d human pose from 2d image landmarks," in *ECCV*, 2012.

[16] S. Avidan and A. Shashua, "Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence," *PAMI*, 2000.

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

[18] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3d trajectory reconstruction under perspective projection," *IJCV*, 2015.

[19] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *CVPR*, 2000.

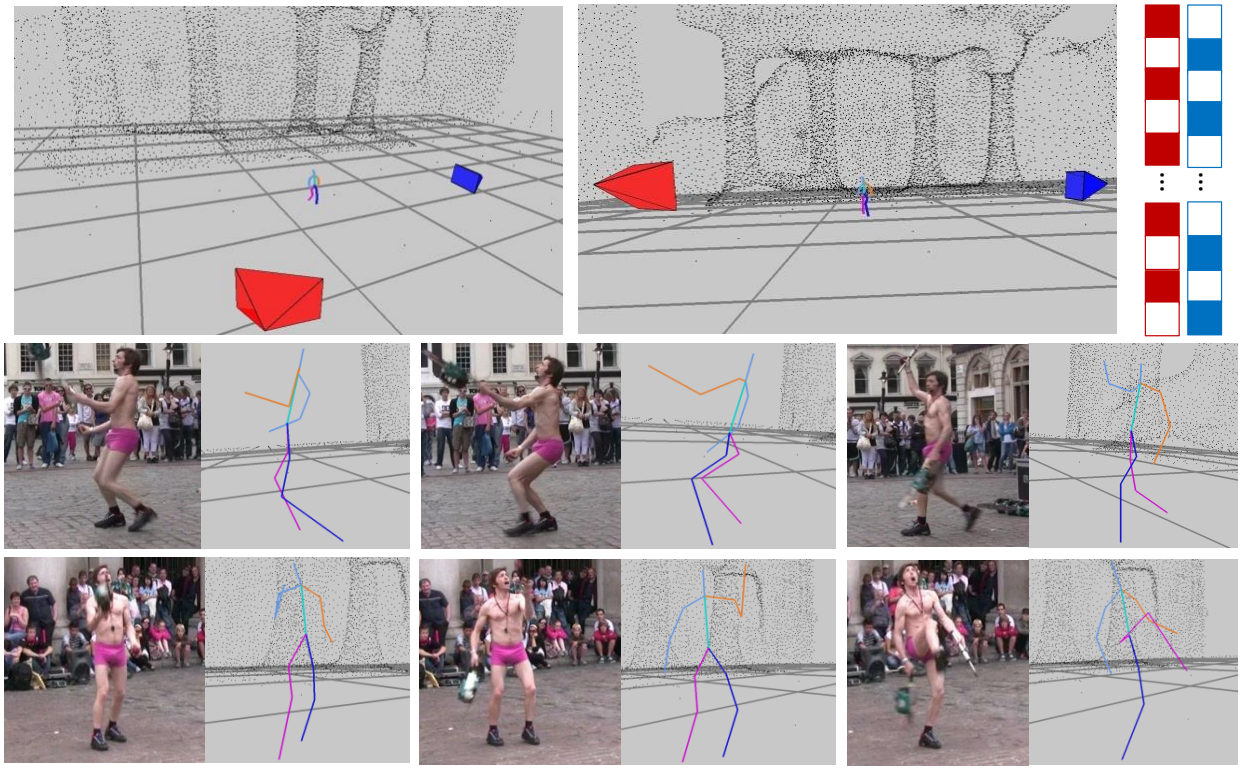
[20] J. Xiao, J. Chai, and T. Kanade, "A closed-form solution to non-rigid shape and motion recovery," in *ECCV*, 2004.

[21] R. Vidal and D. Abretske, "Nonrigid shape and motion from multiple perspective views," in *ECCV*, 2006.

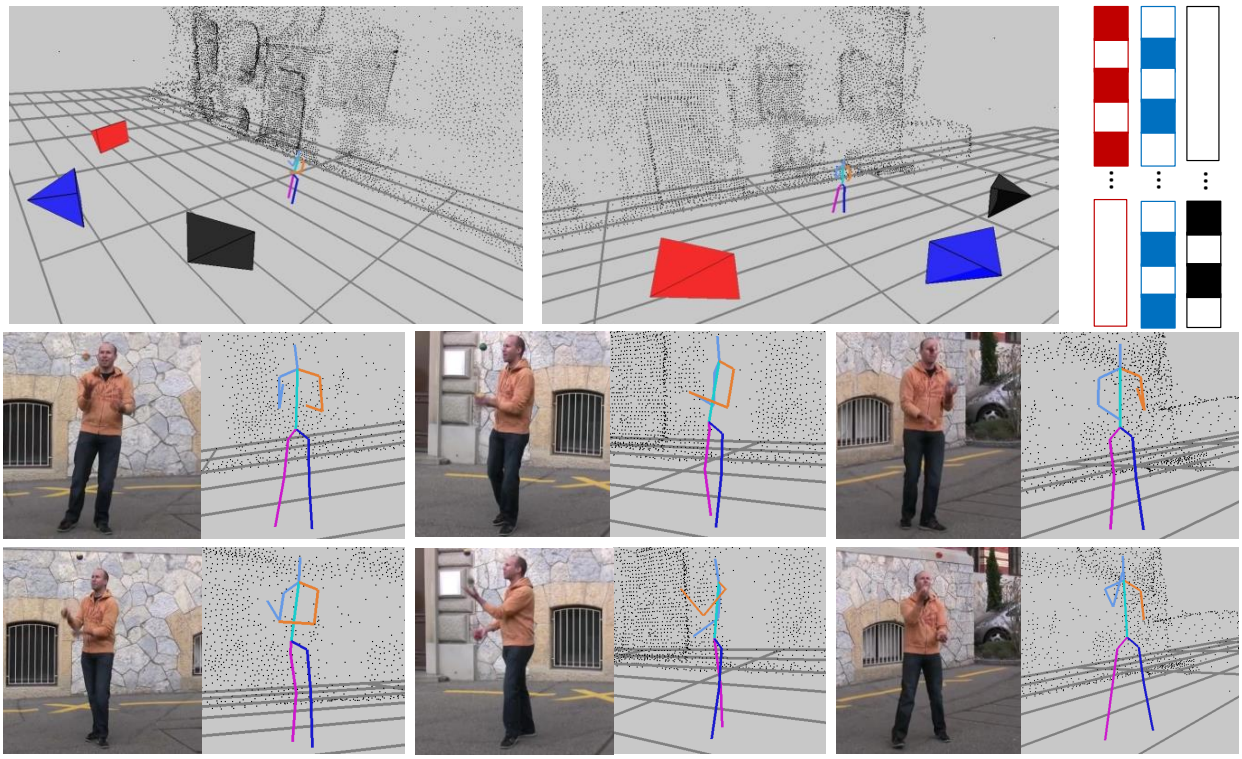
[22] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[24] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient convnet-based



(a) Rothman dataset (250 frames)



(b) Juggler dataset (180 frames)

Fig. 15: The datasets presented in [40]. The frame rate of each camera is 12.5 Hz. For each dataset, the top left two show the camera configuration, the top right describes the temporal distribution of each image sequence (a colored grid means the camera of the same color captures one frame at a time instance), and the bottom shows example reconstruction results. To demonstrate that all the video streams are not necessary to have a common temporal span, in Fig. 15b we choose to let the red and the black camera cover half of the performance.

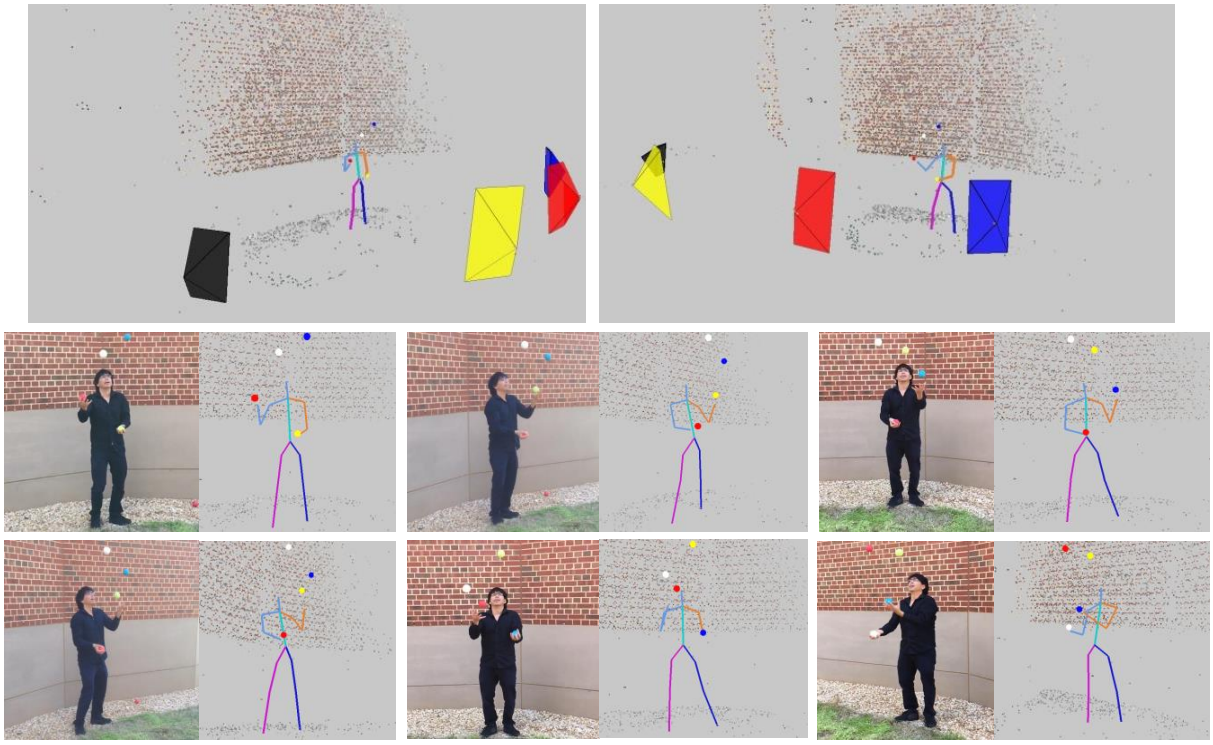


Fig. 16: Results of a person juggling. Note we reconstruct the four juggler balls in addition to the person. The unsynchronized image sequences from iPhone6 and iPhone5 have frame rates of 10 Hz and 6.25 Hz respectively, with unknown temporal distribution.

marker-less motion capture in general scenes with a low number of cameras," in *CVPR*, 2015.

[25] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NIPS*, 2014.

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

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

[28] T. Tuytelaars and L. V. Gool, "Synchronizing video sequences," in *CVPR*, 2004.

[29] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski, "Synchronization of multiple camera videos using audio-visual features," *Multimedia, IEEE Transactions on*, 2010.

[30] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *ICCV*, 2003.

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

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2009.

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

[34] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

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

[36] 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*, 2011.

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

[38] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *PAMI*, 2008.

[39] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *CVPR*, 2009.

[40] L. Ballan, G. Brostow, J. Puwein, and M. Pollefeys, "Unstructured video-based rendering: Interactive exploration of casually captured videos," in *ACM Transactions on Graphics*, 2010.

Enliang Zheng is currently a PhD student in the computer science department of the Univeristy of North Carolina at Chapel Hill. His research interests include static and dynamic object reconstruction, camera pose estimation in structure from motion, image based virtual view synthesis, etc. Before his PhD study, he received the Bachelor and Master degrees from Shandong University and Shanghai Jiaotong University in 2006 and 2009, respectively.

Dinghuang Ji received the BE degree in computer science from the University of Science and Technology of China in 2009, and the MS degree in computer science from the Institute of Computing Technology in 2013. He is currently working toward his PhD degree in Department of Computer Science at UNC Chapel Hill. His research interests include 3D computer vision, image processing and graphics.

Enrique Dunn completed a doctorate in Electronics and Telecommunications in 2006, and a masters degree in Computer Science in 2001, both from the Ensenada Center for Scientific Research and Higher Education, México. He is currently an associate professor at Stevens Institute of Technology. His research interests include structure from motion, dense 3D modeling, large-scale crowd-sourced image analysis, and computational intelligence.

Jan-Michael Frahm received the PhD degree in computer vision in 2005 from the Christian Albrechts University of Kiel, Germany. His diploma in computer science is from the University of Lübeck. He is currently an associate professor at the University of North Carolina at Chapel Hill. His research interests include structure from motion, real-time multiview stereo, robust estimation methods, high-performance feature tracking, and the development of data-parallel algorithms for commodity graphics hardware.

Supplemental Materials: Self-expressive Dictionary Learning for Dynamic 3D Reconstruction

11 The reason our formulation (Eq. (10)) keeps the sparsity-inducing effect.

First we consider the lasso problem given by

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & ||\mathbf{A}\mathbf{w} - \mathbf{b}||_2^2 \\ \text{subject to} \quad & ||\mathbf{w}||_1 \leq a, \end{aligned} \tag{1}$$

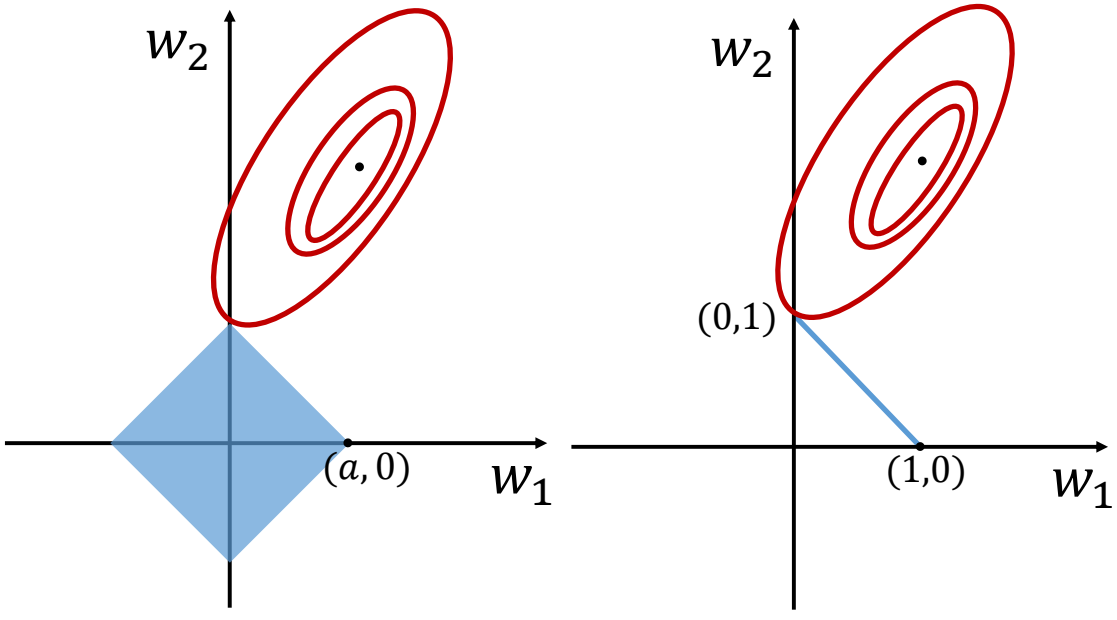
where $\mathbf{A}$ is a matrix, and $\mathbf{w}$ and $\mathbf{b}$ are vectors. Hastie *et al.* [32] describe the intuition behind the mechanism that Eq. (1) is likely to produce sparse results. Fig. 1a depicts lasso when the dimension of vector $\mathbf{w}$ is 2. Denoting $\mathbf{w} = [w_1, w_2]$, the residual $||\mathbf{A}\mathbf{w} - \mathbf{b}||_2^2$ has elliptical contours (level set), and the constraint region for lasso is defined by the diamond $|w_1| + |w_2| \leq a$. Eq. (1) finds the first point where the elliptical contours hit the diamond constraint. If the solution hits the corner, then it has one parameter w_j equal to zero. When $\mathbf{w}$ is in higher dimension, there are more opportunities for the estimated $\mathbf{w}$ to be sparse.

For our formulation, we minimize the problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & ||\mathbf{A}\mathbf{w} - \mathbf{b}||_2^2 \\ \text{subject to} \quad & \mathbf{w} \cdot \mathbf{1} = 1, \\ & \mathbf{w} \geq 0. \end{aligned} \tag{2}$$

In the case that $\mathbf{w}$ has dimension two, we have the constraint that the points have to stay on the line segment (see Fig. 1b), instead of the diamond constraint for the lasso problem. The line segment connects the point (1,0) and (0,1) on the first quadrant. Similar to the lasso problem, the elliptical contours is likely to hit the end of the line segment, hence producing sparse results.

Note the sparsity effect of Eq. (2) is introduced by Chen *et al.* [33], and they also propose an efficient solver for this problem.



(a) lasso (b) our formulation

Fig. 1: Estimation picture of lasso and our formulation