Int J Comput Vis
DOI 10.1007/s11263-015-0804-2

3D Trajectory Reconstruction under Perspective Projection

Hyun Soo Park · Takaaki Shiratori · Iain Matthews · Yaser Sheikh

Received: 18 October 2012 / Accepted: 24 January 2015

Abstract We present an algorithm to reconstruct the 3D trajectory of a moving point from its correspondence in a collection of temporally non-coincidental 2D perspective images, given the time of capture that produced each image and the relative camera poses at each time instant. Triangulation-based solutions do not apply, as multiple views of the point may not exist at each time instant. We represent a 3D trajectory using a linear combination of compact trajectory basis vectors, such as the discrete cosine transform basis, that have been shown to approximate object independence. We note that such basis vectors are also *coordinate* independent, which allows us to directly use camera poses estimated from stationary areas in the scene (in contrast to nonrigid structure from motion techniques where cameras are simultaneously estimated). This reduces the reconstruction optimization to a linear least squares problem, allowing us to robustly handle missing data that often occur due to motion blur, texture deformation, and self occlusion. We present an algorithm to determine the number of trajectory basis vectors, individually for each trajectory via a cross validation scheme and refine the solution by minimizing the geometric error. The relationship between point and camera motion can cause degeneracies to occur. We geometrically analyze the problem by studying the relationship of the camera motion, point motion, and trajectory basis vectors. We define the *reconstructability* of a 3D trajectory under projection, and show that the estimate approaches the ground truth when reconstructability approaches infinity. This analysis enables us to precisely characterize cases when accurate reconstruction is achievable. We present qualitative results for the reconstruction of several real-world scenes from a series of 2D projections where high reconstructability can be guaranteed, and report quantitative results on motion capture sequences.

Keywords Dynamic 3D reconstruction · Trajectory triangulation · Trajectory space · Reconstructability

Communicated by Jun Sato.

H. S. Park (✉) · Y. Sheikh
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: hyunsoop@cs.cmu.edu

Y. Sheikh
e-mail: yaser@cs.cmu.edu

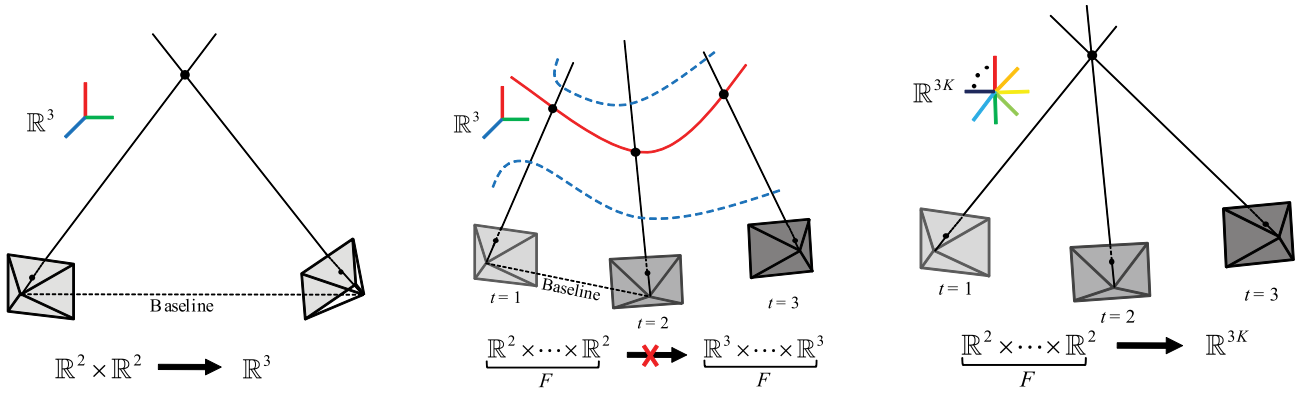
T. Shiratori
Microsoft Research Asia, Beijing, China
e-mail: takaakis@microsoft.com

I. Matthews
Disney Research Pittsburgh, Pittsburgh, USA
e-mail: iainm@disneyresearch.com

1 Introduction

It is impossible to reconstruct a 3D scene from a single image without making prior assumptions about scene structure. Binocular stereoscopy is a solution used by both biological and artificial systems to localize the position of a point in 3D via correspondences in two views. Classic triangulation used in stereo reconstruction is geometrically well-posed, as shown in Fig. 1a. The rays connecting each image location to its corresponding camera center intersect at the true 3D location of the point—this process is called triangulation, as the two rays form a triangle with the baseline that connects the two camera centers. However, the triangulation constraint does not apply when the point moves between image captures, as shown in Fig. 1b. This case abounds as most artificial vision systems are monocular and most real scenes contain moving elements.

Published online: 18 February 2015



(a) Static point (point in $\mathbb{R}^3$) **(b)** Moving point (trajectory in $\mathbb{R}^3$) **(c)** Moving point (point in $\mathbb{R}^{3K}$)

Fig. 1 **a** A 3D point can be triangulated from two or more views; **b** 3D trajectory reconstruction is impossible without any constraint on the trajectory because any trajectory (dotted trajectories) passing through the optical rays can be a solution; **c** We represent a 3D trajectory with a linear combination of compact trajectory basis vectors, which is a point in $\mathbb{R}^{3K}$. This enables us to linearly reconstruct the point trajectory

The 3D reconstruction of a trajectory is directly analogous to monocular image reconstruction. Just as it is impossible to reconstruct a 3D point from a single image without making assumptions about scene structure, it is impossible to reconstruct a moving point without making assumptions about the way it moves. In this paper, we present an algorithm to reconstruct a moving point from a series of 2D perspective projections, given the camera matrix for each projection. We represent the 3D trajectory using a linear combination of compact trajectory basis vectors (Sidenbladh et al. 2000; Akhter et al. 2008, 2011) and demonstrate that, under this model, we can recover 3D point motion via a linear least squares solve. We generalize the problem of 3D point triangulation, which is a mapping from $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$, to 3D trajectory reconstruction, as a mapping $\mathbb{R}^2 \times \cdots \times \mathbb{R}^2 \rightarrow \mathbb{R}^{3K}$. $3K$ is the number of trajectory basis vectors required to represent the 3D point trajectory[1] as shown in Fig. 1c.

Dynamic 3D reconstruction using shape or trajectory basis vectors requires three types of variables to be estimated (Bregler et al. 1999): camera motion, model description (often represented as shape or trajectory basis vectors), and model parameters (often represented as basis coefficients). Simultaneously estimating these three types of parameters results in a trivariate optimization, and constitutes the problem definition of nonrigid structure from motion (NRSfM). The optimization suffers from suboptimality, in general, due to the non-convex objective function, and is sensitive to noise and missing data. Akhter et al. (2008, 2011) reduced the complexity of the trilinear relationship by exploiting the fact that trajectory basis vectors can be object independent and therefore, can be pre-defined. This yielded a bilinear optimization over camera motion and coefficient vectors. In this paper, we note that the pre-defined trajectory basis is also *coordinate* independent—the spectral distribution remains identical under a similarity transform; as a result, we can use stationary points in the scene to separately estimate camera poses using classic structure from motion. Thus, unlike NRSfM, we take cameras estimated by the stationary areas of the scene as input into our algorithm. The resulting optimization can be solved using a linear least squares solve providing stable, accurate, and efficient estimates in the presence of missing data. We demonstrate 3D reconstruction results of dynamic scenes that include whole body motion, multiple interacting people, and activity with significant locomotive displacement.

The stability of classic triangulation is known to depend on the baseline between camera centers (Hartley and Zisserman 2004). We study the instability encountered when interference occurs between the point trajectory and camera trajectory, and characterize the cases when trajectory reconstruction is ambiguous. In particular, we define a criterion called *reconstructability*, a measure of reconstruction accuracy defined by the point trajectory, camera trajectory, and basis vectors. We show that when reconstructability approaches infinity, the obtained solution from least squares approaches the ground truth solution.

Building upon an earlier version of this paper (Park et al. 2010), we present an algorithm to automatically select the number of basis vectors *individually* for each point, and to refine the estimated trajectory. Different points on an object may undergo different degrees of motion. For instance, a point on a hand exhibits much more complicated motion during a walk cycle than a point on the torso. Our algorithm uses a cross validation scheme to independently select the number of basis vectors for each trajectory that defines the degree of motion (motion complexity). We present a nonlinear refine-

[1] Related observations have been made in Shashua and Avidan (2000) and Hartley and Vidal (2008).

ment algorithm that takes the trajectory solution as an initial guess and optimizes the solution based on the geometric reprojection error. We categorize our linear least squares formulation as solvable or unsolvable systems depending on the relationship between camera and point motion, and investigate degenerate cases where the reconstruction is impossible or inaccurate.

Our core contributions include: (1) a linear least squares formulation of 3D trajectory reconstruction, (2) a geometric analysis of the relationship between camera motion, point motion, and basis vectors, and (3) a cross-validation scheme that automatically selects a per-trajectory truncation factor depending on its motion. Our algorithm allows us to reconstruct challenging real-world scenes where there are significant amount of missing data and correspondence noise with perspective cameras. Automatic selection of the basis vectors and the trajectory refinement by geometric error minimization allows the algorithm to handle a wider variety of realistic motion. We identify real-world scenarios where high reconstruction accuracy is guaranteed via the geometric analysis and apply our algorithm to reconstruct the time-varying 3D structure of the scenes.

2 Related Work

Reconstructing a dynamic scene in 3D from a monocular image sequence is fundamentally an ill-posed problem. To overcome its inherent ambiguity, a large body of work has developed algorithms, representations, and scene constraints. There have been two general categories of approaches: trajectory triangulation and nonrigid structure from motion. Our method falls in the first category of methods, but we also relate it to approaches in the latter category. Related discussions on geometric analyses of the problem will also be drawn.

2.1 Trajectory Triangulation

When correspondences are provided across 2D images in static scenes, the method proposed by Longuet-Higgins (1981) estimates the relative camera poses and triangulates the point in 3D using epipolar geometry. In subsequent research, summarized in Faugeras et al. (2001), Ma et al. (2003), and Hartley and Zisserman (2004), the geometry involved in reconstructing 3D scenes has been systematically developed. While a static 3D point can be reconstructed by triangulation as shown in Fig. 1a, when the point moves between the capture of both images, the triangulation method becomes inapplicable; the line segments formed by the baseline and the rays from each camera center to the point no longer form a closed triangle (Fig. 1b).

The principal work in 'triangulating' moving points from a series of images is by Avidan and Shashua (2000), who coined the term *trajectory triangulation*. They demonstrated two cases where a moving point can be reconstructed: (1) if the point moves along a line, or (2) if the point moves along a conic section. This inspired a number of approaches of geometrically constrained trajectory recovery. Shashua and Wolf (2000) and Wexler and Shashua (2000) introduced homography tensors to represent a point moving on the plane. As an integration of the algebraic curve representation, Wolf and Shashua (2002) classified different manifestations of related problems, analyzing them as projections from $\mathbb{P}^N$ to $\mathbb{P}^2$ where N is a factor representing the span of the trajectory space. Kaminski and Teicher (2004) extended these ideas to a 3D trajectory represented by a family of hypersurfaces in the projective space $\mathbb{P}^5$, i.e., a homogeneous polynomial vanishes on the Plücker coordinates of all lines intersecting the trajectory. This method provides a general framework to reconstruct any arbitrary trajectory that can be represented by a polynomial. However, the algorithm is computationally prohibitive and sensitive to noise, which we will discuss in detail in Sect. 5.1.3.

In this paper, we investigate 3D reconstruction of a point trajectory where the point motion can be described as a linear combination of compact trajectory basis vectors. This representation allows far more natural motions to be linearly reconstructed (Akhter et al. 2008, 2011). We demonstrate its application in reconstructing moving points from a series of image projections where no two image projections necessarily occur at the same time instant.

2.2 Nonrigid Structure from Motion

Nonrigid structure from motion is another approach to reconstructing dynamic structure in 3D from a monocular sequence. Unlike the trajectory triangulation approach, nonrigid structure from motion approaches recover camera motion as a part of their optimization. The seminal work of Bregler et al. (1999) introduced linear shape models as a representation for nonrigid 3D structures, and demonstrated their applicability within the factorization-based paradigm of Tomasi and Kanade (1992). They formulated the problem as a trilinear optimization over camera motion, shape basis vectors, and shape coefficient vectors. However, finding a global solution of the trilinear optimization is difficult (Brand 2005; Xiao and Kanade 2004; Xiao et al. 2006; Akhter et al. 2009) because of non-convexity of the objective function. Recent work has considered a number of optimization techniques to overcome the suboptimality issue.Torresani et al. (2001, 2008) used an alternating linear least squares technique and Brand (2001) provided a sophisticated initialization by allowing minimal shape deformation. Paladini et al.

(2009) proposed a robust metric upgrade method by iteratively projecting the solution onto metric motion manifold.

Prior knowledge on shapes that regularizes deformation can improve stability of the optimization. Xiao and Kanade (2004), Xiao et al. (2006) added a shape basis constraint which maximizes the basis independence leading to a closed-form solution. An algorithm to learn shape deformation was introduced by Torresani et al. (2003). Torresani and Bregler (2002) and Olsen and Bartoli (2007) proposed a temporal smoothness prior on the shape basis vectors and camera parameters. Yan and Pollefeys (2005) used an articulation constraint that can limit shape subspace. Del Bue (2008) proposed a pre-computed prior which produces reliable reconstruction when there is degeneracy of motion and Fayad et al. (2010) introduced piecewise reconstruction by dividing the surface into overlapping patches.

When the shape basis vectors are known, the complexity of the trilinear optimization reduces to a bilinear optimization. This complexity reduction results in robust camera motion and shape estimation. A nonrigid structure registration problem given a template is one such domain. Blanz and Vetter (1999) modeled a face using a linear combination of shape basis vectors and registered/manipulated facial deformation given a new face image. A surface is another target structure that has been extensively studied. Salzmann et al. (2007) utilized a low dimensional shape model made of triangle meshes to represent nonrigid surface. Taylor et al. (2010) proposed locally rigid structure from motion by allowing minimal triangular deformation in 3D and Östlund et al. (2012) regularized a deformable surface based on Laplacian matrix of the structure.

In contrast to these methods, which represent the instantaneous shape of an object as a linear combination of basis shape vectors, another line of research (Sidenbladh et al. 2000; Akhter et al. 2008, 2011) modeled each trajectory using a linear combination of trajectory basis vectors. Akhter et al. (Akhter et al. (2008, 2011)) noted that, unlike shape basis approaches, this trajectory representation can be object independent, i.e., it can express any object deformation without prior information, while the shape representation is restricted to the varying shapes of the observed object. This enabled them to use a pre-defined trajectory basis vectors, such as the discrete cosine transform (DCT). The method also reduced the complexity of the trilinear optimization to a bilinear optimization and showed accurate reconstruction when shapes cannot be well modeled by compact shape basis vectors such as articulated motion. Gotardo and Martinez (2011) also used the DCT trajectory basis vectors to handle missing data. Valmadre and Lucey (2012) generalized the trajectory basis concept by formulating the regularization as a temporal filter.

In this paper, we note that the pre-defined trajectory basis vectors can be coordinate independent and therefore camera poses, estimated by stationary points in the scene, can be used directly. This enables us to further reduce the complexity of the original trilinear optimization to a linear optimization where we can find a global solution efficiently—in effect, using the trajectory basis in the 3D *trajectory triangulation problem*. We present a linear solution to reconstruct a moving point from a series of its image projections inspired by the direct linear transform algorithm (Hartley and Zisserman 2004).

Our linear formulation, achieved by known camera poses and the pre-defined trajectory basis vectors, enables us to robustly handle problems like missing data (due to occlusion and matching failure) and estimation instability, which most previous approaches suffer from. The work by Torresani et al. (2008) and Vidal and Hartley (2004) can handle missing data using the rank constraint of the flow matrix. However, all these algorithms remain sensitive to noise and have been demonstrated to work only for largely rigid transformations or constrained deformation of objects, such as faces. The use of stationary areas of the scene to measure camera motion is also related to the approaches proposed by Del Bue et al. (2006) and Bartoli et al. (2008), who use rigid points on the object to estimate the relative camera motion.

Factorization methods largely assume an orthographic camera model while a few methods have used a perspective camera model(Hartley and Vidal 2008; Vidal and Abretske 2006; Zhu et al. 2010; Del Bue et al. 2006; Bartoli et al. 2008; Lladó et al. 2010). We formulate reconstruction under a perspective camera model in terms of linear least squares, which allows us to find the globally optimal solution in the presence of missing data.

2.3 Reconstruction Stability

As noted in the previous section, the optimization for nonrigid structure from motion is known to be difficult because of the non-convex objective function. Xiao et al. (2006) asserted that the orthonormality constraints on camera rotation, which were used in the original factorization framework by Tomasi and Kanade (1992), were not sufficient to disambiguate solutions, and presented additional constraints on the shape basis vectors. Subsequently, Akhter et al. (2009) demonstrated that the orthonormal constraints were theoretically sufficient, i.e., there exists a global solution that can be achieved only by the orthonormality constraint up to rotation and scale. They noted that difficulty to obtain a global solution is not originated from ambiguity but from optimization complexity. Dai et al. (2012) confirmed this claim from their formulation via trace-norm minimization.

Ambiguity of the problem has been widely investigated while the stability of the obtained solution is less under-

stood[2]. Consider two cases: an object deforms in front of a stationary camera and in front of an orbiting camera around the object. The same trilinear optimization algorithm applies but accurate reconstruction cannot be achieved for the first case due to a lack of camera motion, while it is achievable for the second case. The relative camera motion plays a significant role for accurate reconstruction[3]. This is analogous to stationary point triangulation where the baseline that connects two camera centers characterizes the estimation error and its uncertainty in the presence of measurement noise. In this paper, we provide a geometric analysis of the reconstruction problem. As a result, we find that reconstruction accuracy is fundamentally restricted by a criterion we refer to as *reconstructability* that characterizes the relationship between camera motion, point motion, and the trajectory basis.

3 Method

In this section, we present an algorithm to reconstruct the 3D trajectory of a moving point from 2D perspective projections given the 3D camera poses[4] and the time of capture of the cameras. We represent each trajectory using a linear combination of compact trajectory basis vectors and solve for the trajectory coefficient vector via linear least squares in Sect. 3.1. The number of trajectory basis vectors is automatically chosen by a cross validation scheme (Sect. 3.2) and the estimated trajectory is refined by minimizing the geometric error (Sect. 3.3).

3.1 Linear Reconstruction of a 3D Point Trajectory

For a given ith camera projection matrix, $\mathbf{P}_i \in \mathbb{R}^{3\times 4}$, let a point in 3D, $\mathbf{X}_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^\mathsf{T}$, be imaged as $\mathbf{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\mathsf{T}$. The index i represents the ith time sample. This projection is defined up to scale,

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \simeq \mathbf{P}_i \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix}, \text{ or } \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_i \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} = \mathbf{0}, \tag{1}$$

where $[\cdot]_\times$ is the skew symmetric representation of the cross product (Hartley and Zisserman 2004). This can be rewritten as an inhomogeneous equation,

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,1:3}\mathbf{X}_i = -\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,4} \ ,$$

where $\mathbf{P}_{i,1:3}$ and $\mathbf{P}_{i,4}$ are the matrices made of the first three columns and the last column of $\mathbf{P}_i$, respectively, or simply as $\mathbf{Q}_i\mathbf{X}_i = \mathbf{q}_i$, where,

$$\mathbf{Q}_i = \left(\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,1:3} \right)_{1:2}, \ \mathbf{q}_i = -\left(\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,4} \right)_{1:2},$$

and $(\cdot)_{1:2}$ is the matrix made of first two rows from $(\cdot)$. By taking into account all time instances, the 3D point trajectory, $\mathbf{X}$, can be written as,

$$\begin{bmatrix} \mathbf{Q}_1 & & \\ & \ddots & \\ & & \mathbf{Q}_F \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_F \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_F \end{bmatrix}, \text{ or } \mathbf{QX} = \mathbf{q}, \tag{2}$$

where F is the number of time samples in the trajectory.

Since Eq. (2) is an underconstrained system (i.e., $\mathbf{Q} \in \mathbb{R}^{2F\times 3F}$), there are an infinite number of solutions for a given set of measurements (2D projections). We constrain the solution space in which $\mathbf{X}$ lies by approximating the point trajectory using a linear combination of compact trajectory basis vectors,

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} \mathbf{X}_1^\mathsf{T} \cdots \mathbf{X}_F^\mathsf{T} \end{bmatrix}^\mathsf{T} \\ &\approx \boldsymbol{\Theta}_1\beta_1 + \cdots + \boldsymbol{\Theta}_{3K}\beta_{3K} \\ &= \boldsymbol{\Theta}\boldsymbol{\beta}, \end{aligned} \tag{3}$$

where $\boldsymbol{\Theta}_j \in \mathbb{R}^{3F}$ is a trajectory basis vector, $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_1 \ldots \boldsymbol{\Theta}_{3K} \end{bmatrix} \in \mathbb{R}^{3F\times 3K}$ is the trajectory basis matrix, $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \ldots \beta_{3K} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{3K}$ is a trajectory coefficient vector, and K is the number of the trajectory basis vectors per coordinate.

From Eqs. (2) and (3), we can derive the following system of equations,

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

To reconstruct moving points in 3D, we have to solve the following trilinear system (Bregler et al. 1999),

$$\underset{\{\mathbf{P}_i\}_{i=1,\cdots,F},\boldsymbol{\Theta},\boldsymbol{\beta}}{\operatorname{argmin}} \ \|\mathbf{Q}\boldsymbol{\Theta}\boldsymbol{\beta} - \mathbf{q}\|^2, \tag{5}$$

given 2D projections, $\{\mathbf{x}_i\}_{i=1,\cdots,F}$. Akhter et al. (2008) identified that the trajectory basis vectors are *object* independent. This allowed them to use pre-defined trajectory basis vectors such as the DCT and to remove $\boldsymbol{\Theta}$ from the trilinear optimization. This reduced the optimization to a bilinear system,

$$\underset{\{\mathbf{P}_i\}_{i=1,\cdots,F},\boldsymbol{\beta}}{\operatorname{argmin}} \ \|\mathbf{Q}\boldsymbol{\Theta}_{\text{DCT}}\boldsymbol{\beta} - \mathbf{q}\|^2, \tag{6}$$

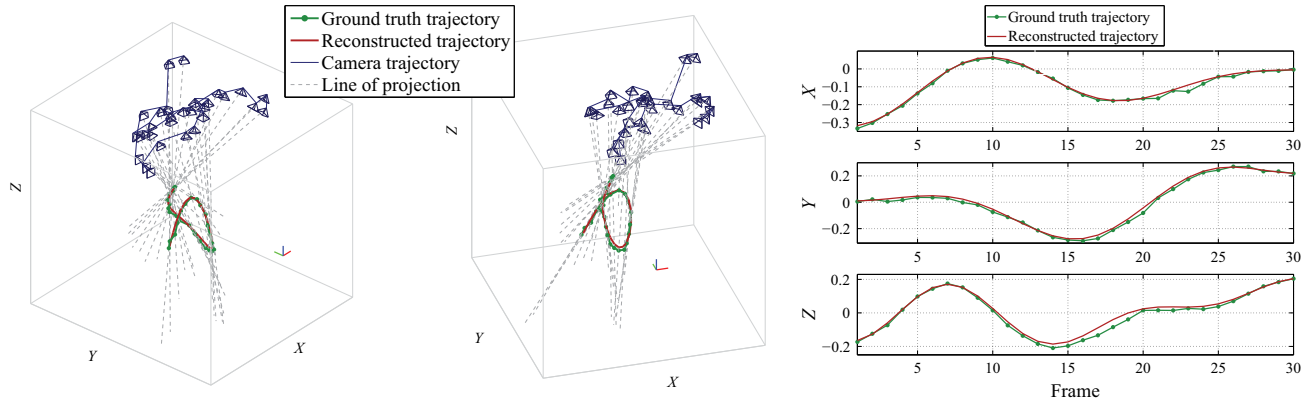
where $\boldsymbol{\Theta}_{\text{DCT}}$ is the pre-defined DCT trajectory basis vectors.

We note that these trajectory basis vectors are also *coordinate* independent, i.e., the trajectory basis vectors can

[2] For the purposes of this discussion, it should be noted that any global rigid motion of the object is equivalent to relative camera motion.

[3] Related empiricial observations have been made by Ozden et al. (2004) and Akhter et al. (2008).

[4] We estimate camera poses automatically via structure from motion. See Sect. 5.2 for a description of the camera pose estimation algorithm.



(a) Trajectory reconstruction from Equation (7)

(b) Comparison with the ground truth

Fig. 2 We reconstruct a trajectory using linear least squares. **a** The reconstructed trajectory is illustrated in two views. The trajectory, which is represented by a linear combination of trajectory basis vectors, passes through all lines of projections. The *blue pyramids* are camera poses. **b** We project the ground truth trajectory and the reconstructed trajectory into the X, Y, and Z axis to show the accuracy of trajectory reconstruction. Trajectory reconstruction via Eq. (7) produces an accurate solution

compactly represent a trajectory equally well in any arbitrary orthogonal world coordinate system via the following Result 1.

Result 1 *The spectral distribution of a 3D trajectory basis is invariant to 3D similarity transforms.*

See the Appendix for a proof. The consequence of Result 1 is that we can estimate the camera motion, $\mathbf{P}_{\text{SfM}}$, independently, using structure from motion on the stationary points in a scene (Hartley and Zisserman 2004) as discussed in Sect. 5.2. This further reduces the bilinear system to a linear system as follows,

$$\underset{\boldsymbol{\beta}}{\text{argmin}} \; \|\mathbf{Q}_{\text{SfM}}\boldsymbol{\Theta}_{\text{DCT}}\boldsymbol{\beta} - \mathbf{q}_{\text{SfM}}\|^2. \tag{7}$$

Solving Eq. (7) for the trajectory coefficient vector, $\boldsymbol{\beta}$, is a linear least squares system if $2F \geq 3K$, which provides an efficient, numerically stable, and globally optimal solution. Figure 2 shows 3D trajectory reconstruction via Eq. (7) in the presence of measurement noise. Figure 2a illustrates the camera trajectory and point trajectory with the lines of projections from two perspectives. The reconstructed trajectory is a trajectory that passes through all lines of projections and that is represented by a linear combination of the trajectory basis vectors. Figure 2b shows how the reconstructed trajectory and ground truth trajectory are similar.

If there are missing data by self-occlusion or measurement noise, the corresponding rows in $\mathbf{Q}$ and $\mathbf{q}$ may be dropped in Eq. (7). As long as the resulting $\mathbf{Q}\boldsymbol{\Theta}$ matrix satisfies the least squares criterion, i.e., $2\widehat{F} > 3K$ where $\widehat{F}$ is the remaining number of measurements, the estimation of $\boldsymbol{\beta}$ is robust. This allows us to handle the problem of missing data.

3.2 Selection of the Number of Basis Vectors

Our approach requires the selection of the number of basis vectors, K. In Akhter et al. (2008, 2011) and Park et al. (2010), the number of the DCT basis vectors was manually tuned and all trajectories were reconstructed with the same number of the basis vectors. This is undesirable because different points may undergo different degrees of motion. The number of the basis vectors controls the complexity of the trajectory motion. For example, a point that undergoes complex motion such as hands in the dance scene shown in Fig. 13a, requires higher K, i.e., high frequency DCT trajectory basis vectors are needed to represent and reconstruct the complex motion; a point that undergoes simple motion such as the left leg can be represented by more aggressive truncation, retaining only low frequency DCT trajectory basis vectors. If K is too high, the algorithm overfits measurement noise, and conversely, if it is too low, the reconstructed trajectory cannot express the detail of the point motion. In this section, we present an approach to automatically select K_i for the ith trajectory rather than manually setting a global value of K. Bartoli et al. (2008) also presented a method to select K for shape basis vectors via coarse-to-fine reconstruction; their approach selects a global truncation factor for all the points, while our method can determine it per point.

To select the number of basis vectors automatically and individually, we use an N-fold cross validation scheme to check the consistency of the reconstructed trajectory. The 2D trajectory is divided into N sets such that each set contains F/N samples that are uniformly distributed in time across the 2D trajectory. When the jth set, $\mathcal{S}_j$, is considered, the reprojection error, e_j, is evaluated from a 3D trajectory reconstructed from the rest of the $N-1$ sets for a given K_i. This is iterated until all N sets are tested. When K_i is too

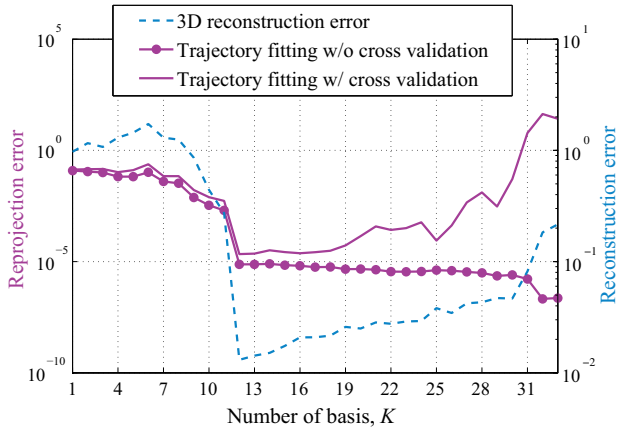


Fig. 3 We select the number of the DCT basis vectors using a cross validation scheme. As the number of the basis vectors, K, increases, reprojection error decreases in general, because a larger K can express the detail of the point trajectory. The *purple line* with markers shows reprojection error as K increases (reprojection error decreases). The *purple line* without markers shows reprojection error measured by our cross validation scheme. When $K = 12$, reprojection error is minimized and the most consistent trajectory through all image measurements is achieved. This also minimizes 3D reconstruction error. Note that the graph has two-sided Y axes, where the left and right Y axes represent reprojection error and 3D reconstruction error in log scale, respectively

high, the trajectory overfits measurement noise, which results in high reprojection error. When K_i is too low, the reprojection error is also high because of limited expressiveness of the basis vectors. We choose the number of the basis vectors for the ith trajectory, which minimizes cross-validated reprojection error, i.e.,

$$K_i^* = \underset{K_i}{\operatorname{argmin}} \sum_{j=1}^{N} e_j(K_i), \tag{8}$$

$$\text{where } K_i = 1, 2, \ldots, \lfloor 2F/3 \rfloor,$$

$$e_j(K_i) = \sum_{s \in \mathcal{S}_j} \left(\frac{\mathbf{P}_s^1 \mathcal{X}_s^{K_i}}{\mathbf{P}_s^3 \mathcal{X}_s^{K_i}} - x_s \right)^2 + \left(\frac{\mathbf{P}_s^2 \mathcal{X}_s^{K_i}}{\mathbf{P}_s^3 \mathcal{X}_s^{K_i}} - y_s \right)^2,$$

$$\mathcal{X}_s^{K_i} = \begin{bmatrix} \boldsymbol{\Theta}(s)^{K_i} \boldsymbol{\beta}^{K_i} \\ 1 \end{bmatrix},$$

where $\lfloor \cdot \rfloor$ is the floor operator (the largest integer not greater than $\cdot$). $\boldsymbol{\Theta}(s)^{K_i} \in \mathbb{R}^{3 \times 3K_i}$ is the trajectory basis vectors evaluated at the sth time instant with the $3K_i$ trajectory basis vectors and $\mathbf{P}^l$ is the lth row of the matrix $\mathbf{P}$. x_s and y_s are a 2D measurement at the sth time instant. In Fig. 3, the purple line with markers plots the reprojection error as K_i increases. The higher the truncation factor K_i, the lower the reprojection error because the details of the trajectory can be expressed. However, a higher K_i may overfit measurement noise of the trajectory. From our cross validation scheme, we are able to automatically select K_i that is the most expressible but the least overfitted. The purple line without markers shows reprojection error and $K_i = 12$ produces the most consistent trajectory for all image measurements (minimum reprojection error) in the presence of measurement noise. This K_i also minimizes 3D reconstruction error.

3.3 3D Trajectory Refinement

Trajectory reconstruction from Eq. (7) minimizes the algebraic error (Hartley and Zisserman 2004). The solution, $\boldsymbol{\beta}$, is not necessarily the maximum likelihood solution under Gaussian measurement noise. We refine the linearly reconstructed trajectory by minimizing the reprojection error, i.e.,

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{F} \left(\frac{\mathbf{P}_i^1 \mathcal{X}_i}{\mathbf{P}_i^3 \mathcal{X}_i} - x_i \right)^2 + \left(\frac{\mathbf{P}_i^2 \mathcal{X}_i}{\mathbf{P}_i^3 \mathcal{X}_i} - y_i \right)^2, \tag{9}$$

$$\text{where } \mathcal{X}_i = \begin{bmatrix} \boldsymbol{\Theta}(i)\boldsymbol{\beta} \\ 1 \end{bmatrix},$$

$\boldsymbol{\Theta}(i) \in \mathbb{R}^{3 \times 3K}$ is the trajectory basis vectors evaluated at ith time instant.

4 Geometric Analysis of 3D Trajectory Reconstruction

In practice, we observe the point trajectory reconstruction approaches the ground truth if the camera motion in relation to the point motion is sufficiently large when the DCT trajectory basis vectors are used. Conversely, if the camera motion is small in relation to the point motion, the solution tends to deviate from the ground truth (Ozden et al. 2004; Akhter et al. 2009). In this section, we analyze the stability of trajectory reconstruction from Eq. (7) by considering the geometric relationship between the point trajectory, the camera center trajectory, and the trajectory basis vectors. We categorize trajectory reconstruction as a solvable or unsolvable system. Trajectory reconstruction is possible only when our least squares system is solvable. More importantly, a solvable system does not guarantee an accurate estimate of the trajectory parameters. We define a measure of reconstruction accuracy, *reconstructability*, for solvable systems. Reconstructability enables us to precisely characterize when accurate reconstruction of a 3D trajectory is possible.[5]

[5] Ambiguity analyses have been investigated by Xiao et al. (2006), Vidal and Abretske (2006), Hartley and Vidal (2008), and Akhter et al. (2009). However, these analyses consider the ambiguity with the use of a shape basis representation, which utilizes the correlation across *multiple* points. In this section, we consider the case of the reconstruction of a *single* 3D point trajectory.

4.1 Geometry of Camera Trajectory, Point Trajectory, and Trajectory Basis Vectors

Let $\mathbf{X}$ and $\widehat{\mathbf{X}}$ be a ground truth trajectory and an estimated point trajectory, respectively. The camera matrix can, without loss of generality, be normalized by intrinsic and rotation matrices, $\mathbf{K}$ and $\mathbf{R}$, respectively, (as all camera matrices are known), i.e., $\mathbf{R}_i^{\mathsf{T}}\mathbf{K}_i^{-1}\mathbf{P}_i = \left[\mathbf{I}_3 \middle| -\mathbf{C}_i\right]$, where $\mathbf{P}_i = \mathbf{K}_i\mathbf{R}_i\left[\mathbf{I}_3 \middle| -\mathbf{C}_i\right]$, $\mathbf{C}_i$ is the camera center, and $\mathbf{I}_3$ is a 3×3 identity matrix. This follows from the fact that triangulation and 3D trajectory reconstruction are both geometrically unaffected by the rotation of the camera about its center. All $\mathbf{P}_i$ subsequently used in this analysis are normalized camera matrices, i.e., $\mathbf{P}_i = \left[\mathbf{I}_3 \middle| -\mathbf{C}_i\right]$. Then, a measurement is a projection of $\mathbf{X}_i$ onto the image plane from Eq. (1). Since Eq. (1) is defined up to scale, the measurement, $\mathbf{x}_i$, can be replaced as follows,

$$\left[\mathbf{P}_i \begin{bmatrix}\mathbf{X}_i \\ 1\end{bmatrix}\right]_{\times} \mathbf{P}_i \begin{bmatrix}\widehat{\mathbf{X}}_i \\ 1\end{bmatrix} = 0. \tag{10}$$

Inserting $\mathbf{P}_i = \left[\mathbf{I}_3 \middle| -\mathbf{C}_i\right]$ results in,

$$[\mathbf{X}_i - \mathbf{C}_i]_{\times}\left(\widehat{\mathbf{X}}_i - \mathbf{C}_i\right) = 0, \tag{11}$$

or equivalently,

$$[\mathbf{X}_i - \mathbf{C}_i]_{\times}\,\widehat{\mathbf{X}}_i = [\mathbf{X}_i]_{\times}\,\mathbf{C}_i. \tag{12}$$

To satisfy Eq. (12), $\widehat{\mathbf{X}}_i$ has to lie in the space spanned by $\mathbf{X}_i$ and $\mathbf{C}_i$, or $\widehat{\mathbf{X}}_i = a_1\mathbf{X}_i + a_2\mathbf{C}_i$. It can be easily verified that $a_2 = 1 - a_1$ by substituting in Eq. (12). Thus, the solution of Eq. (12) is,

$$\widehat{\mathbf{X}}_i = a_i\mathbf{X}_i + (1 - a_i)\mathbf{C}_i, \tag{13}$$

where a_i is an arbitrary scalar. Geometrically, Eq. (13) is a constraint for the perspective camera model that enforces the solution to lie on the ray joining the camera center and the point in 3D. By generalizing the ith point to a point trajectory, Eq. (13) becomes,

$$\widehat{\mathbf{X}} = \mathbf{A}\mathbf{X} + (\mathbf{I} - \mathbf{A})\mathbf{C}, \tag{14}$$

where $\mathbf{A} = \mathbf{D} \otimes \mathbf{I}_3$[6].

From Eq. (3), Eq. (14) can be rewritten as $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} \approx \mathbf{A}\mathbf{X} + (\mathbf{I} - \mathbf{A})\mathbf{C}$.

Figure 4 illustrates the geometry of the solution of Eq. (7). Let the subspace, p, be the space spanned by the trajectory basis vectors, col($\boldsymbol{\Theta}$). The solution $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$, has to simultaneously lie in l and col($\boldsymbol{\Theta}$) where l is a hyperplane that contains the camera trajectory and the point trajectory. Thus, $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$ is the intersection of the hyperplane l and the subspace p. Note that the line and the plane are a conceptual 3D vector space representation for the $3F$-dimensional space. The camera center trajectory, $\mathbf{C} = \left[\mathbf{C}_1^{\mathsf{T}} \ldots \mathbf{C}_F^{\mathsf{T}}\right]^{\mathsf{T}}$, and the point trajectory, $\mathbf{X}$, are projected onto col($\boldsymbol{\Theta}$) as $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}$ and $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$, respectively.

[6] $\otimes$ is the Kronecker product and $\mathbf{D}$ is a diagonal matrix which consists of $\{a_1, \ldots, a_F\}$, the scalar for each point along the trajectory.

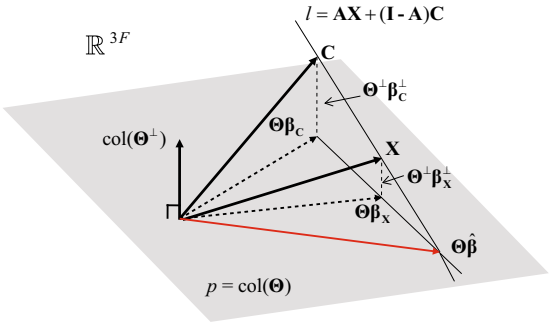


Fig. 4 Geometric illustration of the least squares solution of Eq. (7). The trajectory $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$ is placed at the intersection between the hyperplane l containing the camera trajectory space and the point trajectory, and the p space spanned by the trajectory basis vectors, col($\boldsymbol{\Theta}$)

4.2 Characterization of Trajectory Reconstruction

Recovering $\boldsymbol{\beta}$ depends on the camera trajectory as shown in Fig. 4. We study the degeneracy of the solution of Eq. (7) to characterize the cases when trajectory reconstruction is impossible. The least squares system of Eq. (7) is solvable if rank($\mathbf{Q}\boldsymbol{\Theta}$) $= 3K$ (i.e., it has full column rank).

4.2.1 Unsolvable Systems

When the system is unsolvable, there is a space of solutions where trajectory estimation is ambiguous. We characterize such unsolvable systems as follows,

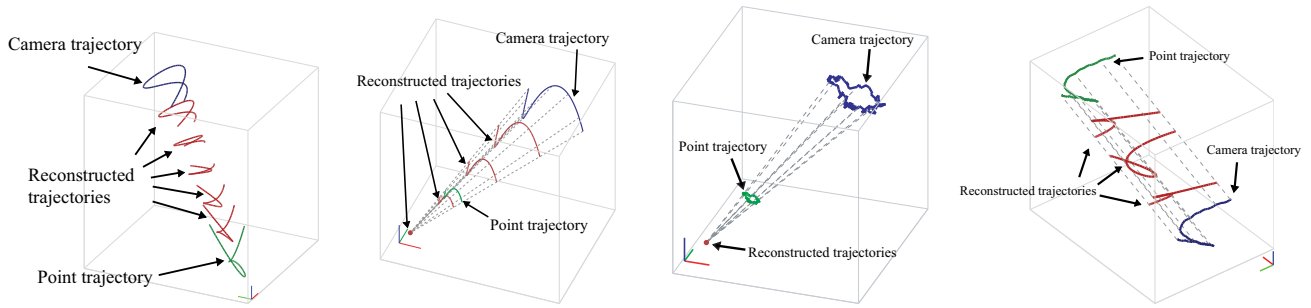
Result 2 Trajectory reconstruction via Eq. (7) is unsolvable if

(i) $\mathbf{X}, \mathbf{C} \in$ col($\boldsymbol{\Theta}$), or

(ii) $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ where c is a nonzero scalar, $\mathbf{1}$ is an F dimensional vector whose entries are all ones, and $\mathbf{d} \in \mathbb{R}^3$ is an arbitrary vector.

Proof (i) If $\mathbf{X}, \mathbf{C} \in$ col($\boldsymbol{\Theta}$), $\mathbf{X} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$ and $\mathbf{C} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}$. Then,

$$\text{null}(\mathbf{Q}\boldsymbol{\Theta}) = \text{null}\left(\begin{bmatrix}[\mathbf{X}_1 - \mathbf{C}_1]_{\times} & & \\ & \ddots & \\ & & [\mathbf{X}_F - \mathbf{C}_F]_{\times}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_F\end{bmatrix}\right)$$



(a) $\mathbf{C}, \mathbf{X} \in \text{col}(\boldsymbol{\Theta})$ **(b)** $\mathbf{C}, \mathbf{X} \in \text{col}(\boldsymbol{\Theta})$ and $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ **(c)** $\mathbf{C}, \mathbf{X} \notin \text{col}(\boldsymbol{\Theta})$ and $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ **(d)** $\mathbf{C}, \mathbf{X} \notin \text{col}(\boldsymbol{\Theta})$ and $\mathbf{X} = \mathbf{C} + \mathbf{1} \otimes \mathbf{d}$

Fig. 5 We illustrate unsolvable systems that produce an infinite number of solutions or a trivial solution. **a** Trajectory reconstruction is ambiguous when $\mathbf{C}, \mathbf{X} \in \text{col}(\boldsymbol{\Theta})$ because there exists $\text{null}(\mathbf{Q}\boldsymbol{\Theta})$, which is an unsolvable system. Plausible reconstructed trajectories that satisfy Eq. (7) are illustrated. **b** Plausible reconstructed trajectories that satisfy Eq. (7) when $\mathbf{C}, \mathbf{X} \in \text{col}(\boldsymbol{\Theta})$ and $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ are shown. **c** When $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ where $c \neq 1$, the solution of the system is always $\mathbf{1} \otimes \mathbf{d}/(1-c)$, which is trivial. **d** When $\mathbf{X} = \mathbf{C} + \mathbf{1} \otimes \mathbf{d}$, the system is unsolvable because $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) = 2K$

$$
\begin{aligned}
&= \text{null}\left(\begin{bmatrix} [\boldsymbol{\Phi}_1(\boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C})]_\times & & \\ & \ddots & \\ & & [\boldsymbol{\Phi}_F(\boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C})]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_F \end{bmatrix}\right) \\
&= \text{null}\left(\begin{bmatrix} [\boldsymbol{\Phi}_1(\boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C})]_\times \boldsymbol{\Phi}_1 \\ \vdots \\ [\boldsymbol{\Phi}_F(\boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C})]_\times \boldsymbol{\Phi}_F \end{bmatrix}\right) \\
&\ni \boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C}, \qquad (15)
\end{aligned}
$$

where $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Phi}_1^\mathsf{T} \cdots \boldsymbol{\Phi}_F^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Since there exists a null space of $\mathbf{Q}\boldsymbol{\Theta}$, $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) < 3K$.

(*ii*) Let us consider two cases where $c \neq 1$ and $c = 1$.

When $c \neq 1$, by plugging $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ into the Eq. (12), it becomes,

$$
\begin{aligned}
[(c-1)\mathbf{C}_i + \mathbf{d}]_\times \widehat{\mathbf{X}}_i &= [c\mathbf{C}_i + \mathbf{d}]_\times \mathbf{C}_i \\
&= [\mathbf{d}]_\times \mathbf{C}_i. \qquad (16)
\end{aligned}
$$

From Eq. (16), $\widehat{\mathbf{X}}_i = \alpha\mathbf{C}_i + (1-\alpha)\mathbf{d}/(1-c)$ where α is a scalar. When $\mathbf{C} \in \text{col}(\boldsymbol{\Theta})$, it is the case where the first condition (*i*) holds, where the system is unsolvable. When $\mathbf{C} \notin \text{col}(\boldsymbol{\Theta})$, $\alpha = 0$ because any component of $\mathbf{C}$ that cannot be expressed by the trajectory basis vectors results in the residual error of Eq. (3). Only $\mathbf{1} \otimes \mathbf{d}/(1-c)$ nullifies the residual error of Eq. (7) but it is still a trivial solution (i.e., a reconstructed trajectory, $\widehat{\mathbf{X}} = \mathbf{1} \otimes \mathbf{d}/(1-c)$, is simply a stationary point even though the point undergoes motion.).

When $c = 1$, $\mathbf{d}/(1-c)$ term in $\widehat{\mathbf{X}}_i = \alpha\mathbf{C}_i + (1-\alpha)\mathbf{d}/(1-c)$ is indeterminate. It is the case where the camera moves exactly the same way the point moves with some offset and $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) = 2K$ because from Eq. (12) and $\mathbf{X} = \mathbf{C} + \mathbf{1} \otimes \mathbf{d}$,

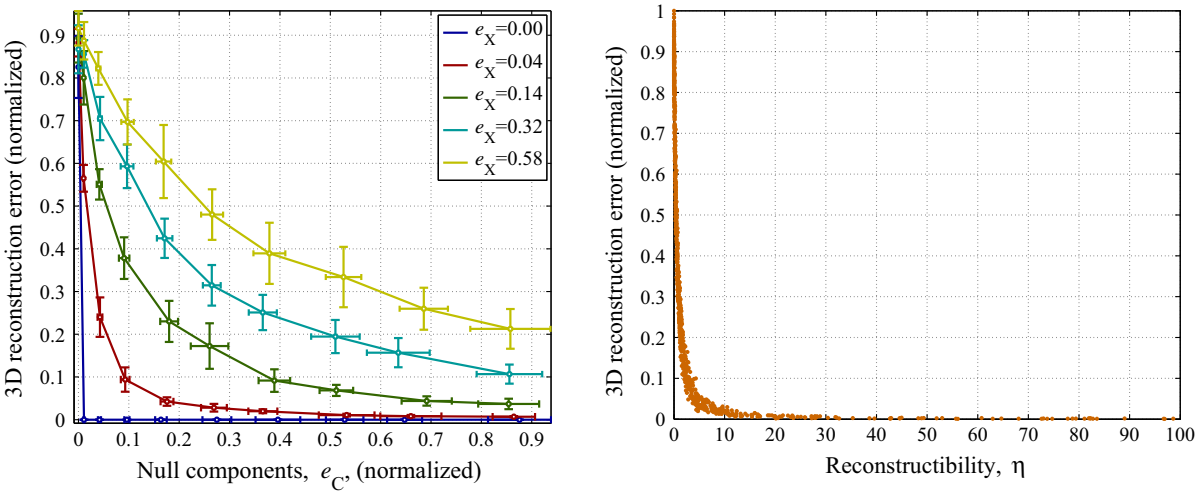
$$
\begin{aligned}
&\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) \\
&= \text{rank}\left(\begin{bmatrix} [\mathbf{d}]_\times & & \\ & \ddots & \\ & & [\mathbf{d}]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_F \end{bmatrix}\right) \\
&= \text{rank}\left(\begin{bmatrix} \mathbf{0} & -d_3\theta_1 & d_2\theta_1 \\ d_3\theta_1 & \mathbf{0} & -d_1\theta_1 \\ \vdots & \vdots & \vdots \\ \mathbf{0} & -d_3\theta_F & d_2\theta_F \\ d_3\theta_F & \mathbf{0} & -d_1\theta_F \end{bmatrix}\right) \\
&= \text{rank}\left(\begin{bmatrix} \mathbf{0} & -d_3\theta_1 & d_2\theta_1 \\ \vdots & \vdots & \vdots \\ \mathbf{0} & -d_3\theta_F & d_2\theta_F \end{bmatrix}\right) + \text{rank}\left(\begin{bmatrix} d_3\theta_1 & \mathbf{0} & -d_1\theta_1 \\ \vdots & \vdots & \vdots \\ d_3\theta_F & \mathbf{0} & -d_1\theta_F \end{bmatrix}\right) \\
&= 2K,
\end{aligned}
$$

where $\mathbf{d} = \begin{bmatrix} d_1 \; d_2 \; d_3 \end{bmatrix}^\mathsf{T}$ and $\boldsymbol{\Phi}_i = \text{blkdiag}\{\theta_i, \theta_i, \theta_i\}$. The trajectory basis vectors for each coordinate (x, y, and z) are the same. Since the rank of the system is $2K$, the system is unsolvable. □

Figure 5 illustrates solutions of unsolvable systems. For Result 2.*i*, Fig. 5a shows an ambiguous solution of Eq. (7) when $\mathbf{X}, \mathbf{C} \in \text{col}(\boldsymbol{\Theta})$. All reconstructed trajectories lie in one dimensional subspace $\boldsymbol{\beta}_\mathbf{X} - \boldsymbol{\beta}_\mathbf{C}$. When $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ (i.e., Result 2.*ii*), the system is also unsolvable. When $c \neq 1$, the solution is $\alpha\mathbf{C}_i + (1-\alpha)/(1-c)\mathbf{d}$. α can be nonzero only when $\mathbf{C} \in \text{col}(\boldsymbol{\Theta})$. Figure 5b illustrates the space of solutions by varying α. When $\mathbf{C} \notin \text{col}(\boldsymbol{\Theta})$, $\alpha = 0$ and the solution is always $\mathbf{1} \otimes \mathbf{d}/(1-c)$ (i.e., stationary point) which is a trivial solution as shown in Fig. 5c. Figure 5(d) shows trajectory reconstruction when $c = 1$, which results in $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) = 2K$. Any trajectory in K dimensional subspace (i.e., $\text{null}(\mathbf{Q}\boldsymbol{\Theta})$) is a solution lying on a surface made by the point trajectory and the camera trajectory, which is shown by gray dotted lines.

4.2.2 Solvable Systems

Result 2 considers an unsolvable system or a system resulting in a trivial solution. For a solvable system, Eq. (7) can



(a) Orthogonality **(b)** Reconstructability

Fig. 6 **a** As the null component of the camera trajectory $e_{\mathbf{C}}$ decreases, the solution of Eq. (7) deviates from the ground truth. **b** The reconstructability η provides the degree of interference between the camera trajectory and point trajectory. Reconstructability is inversely proportional to 3D reconstruction error

be solved without ambiguity in a least squares sense and there exists a unique solution, $\widehat{\boldsymbol{\beta}}$. However, the solvable system does not guarantee an accurate solution: how much $\widehat{\boldsymbol{\beta}}$ deviates from $\boldsymbol{\beta}_{\mathbf{X}}$. We observe that the accuracy of trajectory reconstruction depends on the relationship between the camera center's 3D trajectory, the 3D point trajectory, and the trajectory basis vectors. Given this observation, we characterize the case when reconstruction is accurate.

Solving the least squares system, $\widehat{\mathbf{X}} = \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$, minimizes the residual error by Eq. (14),

$$\underset{\widehat{\boldsymbol{\beta}},\mathbf{A}}{\operatorname{argmin}} \left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\mathbf{X} - (\mathbf{I}-\mathbf{A})\,\mathbf{C} \right\|^2 . \tag{17}$$

Let us decompose the point trajectory and the camera trajectory into the column space of $\boldsymbol{\Theta}$ and that of the null space, $\boldsymbol{\Theta}^{\perp}$ as follows, $\mathbf{X} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} + \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}$, $\mathbf{C} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} + \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}$, where $\boldsymbol{\beta}^{\perp}$ is the coefficient vector for the null space. Let us also define a measure of *reconstructability*, η, of the 3D point trajectory reconstruction,

$$\eta\left(\boldsymbol{\Theta}\right) = \frac{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\|}{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\|} \simeq \frac{\text{How poorly } \boldsymbol{\Theta} \text{ describes } \mathbf{C}}{\text{How poorly } \boldsymbol{\Theta} \text{ describes } \mathbf{X}}. \tag{18}$$

Reconstructability enables us to define the accuracy of the trajectory reconstruction by the following result.

Result 3 $\lim_{\eta\to\infty} \widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{\mathbf{X}}$.

Proof From the triangle inequality, the square root of the objective function of Eq. (17) is bounded by (when $\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\| \to 0$),

$$\begin{aligned}
&\left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} - \mathbf{A}\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp} \right. \\
&\left. \quad - (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp} \right\| \qquad (19)
\end{aligned}$$

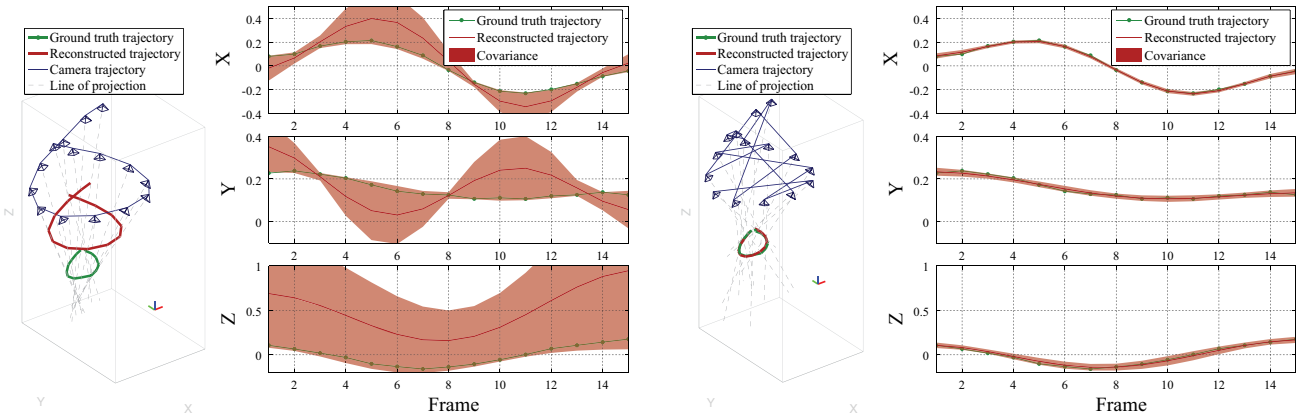
$$\begin{aligned}
&\leq \left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} \right\| + \left\| \mathbf{A}\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp} \right\| \\
&\quad + \left\| (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp} \right\| \\
&\leq \left\| \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp} \right\| \left(\frac{\left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} \right\|}{\left\| \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp} \right\|} \right. \\
&\quad \left. + \frac{\|\mathbf{A}\|}{\eta} + \|\mathbf{I}-\mathbf{A}\| \right), \qquad (20)
\end{aligned}$$

or when $\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\| \to \infty$,

$$\begin{aligned}
&\leq \left\| \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp} \right\| \left(\frac{\left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - (\mathbf{I}-\mathbf{A})\,\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} \right\|}{\left\| \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp} \right\|} \right. \\
&\quad \left. + \|\mathbf{A}\| + \|\mathbf{I}-\mathbf{A}\|\,\eta \right). \qquad (21)
\end{aligned}$$

As η approaches infinity, $\|\mathbf{A}\|/\eta$ in Eq. (20) becomes zero or $\|\mathbf{I}-\mathbf{A}\|\,\eta$ in Eq. (21) becomes infinity. To minimize either Eq. (20) or Eq. (21), $\mathbf{A} = \mathbf{I}$ because it leaves the last term zero and $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{\mathbf{X}}$ because it cancels the first term. This causes the minimum of Eq. (20) or Eq. (21) to become zero, which upper-bounds the minimum of Eq. (19). Thus, as η approaches infinity, $\widehat{\boldsymbol{\beta}}$ approaches $\boldsymbol{\beta}_{\mathbf{X}}$. □

Figure 6a shows how reconstructability is related to the accuracy of the 3D reconstruction error. In each reconstruction, the residual error (null components) of the point trajectory, $e_{\mathbf{X}} = \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\|$, and the camera trajectory, $e_{\mathbf{C}} = \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\|$, are measured. Increasing $e_{\mathbf{C}}$ for a given point trajectory enhances the accuracy of the 3D reconstruction, while increasing $e_{\mathbf{X}}$ lowers accuracy. Even though we cannot directly measure the reconstructability (we do not know the true point trajectory in a real example), it is useful to understand the direct relation with 3D reconstruction accuracy. Figure 6b illustrates that the reconstructability is inversely proportional to the 3D reconstruction error.



(a) Low reconstructability

(b) High reconstructability

Fig. 7 The stability of trajectory reconstruction depends on reconstructability. We reconstruct the same point trajectory with the same camera location but with different ordering. Note that the DCT trajectory basis vectors are used for reconstruction. **a** The order of capture forms a smooth camera trajectory (*left column*), which results in low reconstructability ($\eta = 0.77$) as the camera trajectory can be well represented by the DCT trajectory basis vectors. The reconstructed point trajectory is inaccurate and the covariance of the trajectory is large (*right column*). **b** We shuffle the order of capture to produces a camera trajectory that cannot be well spanned by the DCT trajectory basis vectors while the camera poses remain the same (see the camera trajectory on *left column*). This results in high reconstructability ($\eta = 54.78$). The reconstructed point trajectory is accurate and the covariance of the trajectory is small (*right column*)

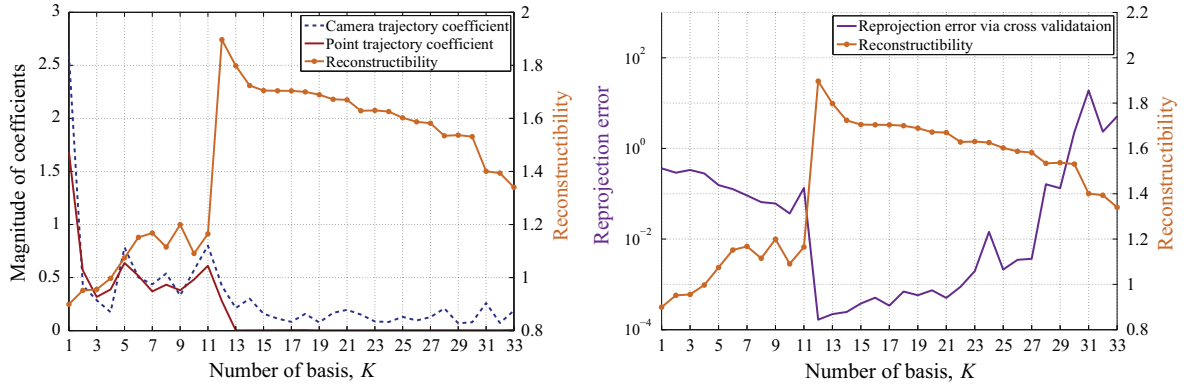
4.2.3 Discussion on Reconstructability

Reconstructability provides key insights into the fundamental relationship between the camera trajectory, point trajectory, and trajectory basis vectors for trajectory reconstruction in 3D and it explains why certain types of the camera motion produce high 3D reconstruction error. Although the number of views is sufficiently large to overconstrain the linear systems in Eq. (7), the reconstructed trajectory can deviate from the ground truth trajectory due to interference of the camera motion. This explains our earlier observations: when the DCT trajectory basis vectors are used, larger camera motion produces larger null components of the camera trajectory, i.e., $\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\|$, which increases reconstructibility according to Eq. (18), and vice versa. Similar observations can be also found in Akhter et al. (2008). As a truncated DCT basis essentially encode smooth motion, by retaining the low frequencies of motion, the basis usually represents the motion of natural points well, but it is likely to capture the motion of cameras equally well. This is a key limitation exposed by our analysis in representing a trajectory using a linear combination of trajectory basis vectors. An important direction of future work is to leverage the knowledge of camera motion to design an optimal subspace to capture low frequency motion.

Reconstructibility is analogous to the baseline which connects two camera centers in classic triangulation as shown Fig. 1a. Stability or uncertainty of point reconstruction is dependent on the baseline between camera centers. If the baseline is wide, the uncertainty of the 3D reconstructed point is small and the stability of estimation is high. If the baseline is narrow, reconstructing the point is highly unstable (i.e., high uncertainty along the rays of projections) in the presence of Gaussian noise. Thus, the baseline provides a key insight of the stability of the reconstruction. Reconstructability is the corresponding concept of the baseline for nonrigid structure from motion in trajectory space.

Figure 7 illustrates how reconstructability relates with reconstruction accuracy and the covariance of the reconstructed trajectory when the DCT trajectory basis vectors are used. We generate a smooth camera trajectory and point trajectory as shown in the left column of Fig. 7a. The smooth camera trajectory forms low reconstructability ($\eta = 0.77$) as the camera trajectory can be well represented by the DCT trajectory basis vectors. Trajectory reconstruction is inaccurate and the covariance of the reconstructed trajectory is large (the right column of Fig. 7a). In Fig. 7b, we shuffle the order of capture to produce a camera trajectory that cannot be well spanned by the DCT trajectory basis vectors while the camera poses remain the same in Fig. 7a. Note that the locations of the camera centers are the same but the camera motion is larger in the left column of Fig. 7b. The large camera motion results in high reconstructability ($\eta = 54.78$). This camera trajectory reconstructs the accurate point trajectory with low covariance as shown in the right column of Fig. 7b.

In practice, the infinite reconstructability criterion is difficult to satisfy because the actual $\mathbf{X}$ is unknown. To enhance reconstructability we can maximize $e_{\mathbf{C}}$ with constant $e_{\mathbf{X}}$. Thus, the best camera trajectory for a given trajectory basis matrix is the one that lives in the null space, col($\boldsymbol{\Theta}^{\perp}$). This explains our observation about small and large camera motion described at the beginning of this section. When the camera motion is small comparing to the point motion, the



(a) Reconstructability and trajectory spectrum

(b) Reconstructability and reprojection error

Fig. 8 Reconstructability and the cross validation scheme are highly related; when reconstructability is maximized, the reprojection error used for the cross validation is minimized. **a** The magnitude of coefficient vectors of the point and camera trajectories is plotted and reconstructability when K basis vectors are used is overlaid. Reconstructability is maximized when the magnitude of coefficients of the point trajectory is diminished ($K = 12$). **b** Reprojection error for the cross validation is minimized where reconstructability is maximized ($K^* = 12$) because that number of basis vectors is the most expressible and the least overfitted

camera trajectory is likely to be represented well by the DCT basis vectors, which results in low reconstructability and vice versa. However, for a given camera trajectory, there is no deterministic way to define trajectory basis vectors because it is coupled with both the camera trajectory and the point trajectory. If one simply finds orthogonal space to the camera trajectory, in general, it is likely to nullify space that also spans the point trajectory space. Geometrically, simply changing the orientation of p in Fig. 4 may result in a greater deviation between $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$ and $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$.

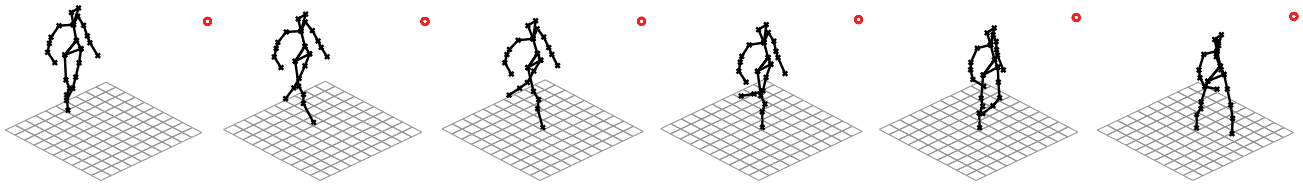
Reconstructability is highly related to the selection of the number of basis vectors via our cross validation scheme described in Sect. 3.2. Given camera motion, reconstructability varies as the number of basis vectors changes as shown in Fig. 8. Figure 8a shows the relationship between the magnitude of the coefficient vectors used to reconstruct the point and camera trajectories, and the reconstructability principle. The selected $K^* = 12$ is the minimum number of trajectory basis vectors that also minimizes the 3D reconstruction error. K^* is the automatically selected number of basis vectors via the cross validation scheme. When $K < K^*$, $||\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}||$ is not minimized because there are some coefficients at higher than the K frequency. When $K > K^*$, $||\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}||$ is already minimized but $||\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}||$ is not maximized. When $K = K^*$, reconstructability is maximized and reprojection error that is used for the cross validation is simultaneously minimized as shown in Fig. 8b.

5 Results

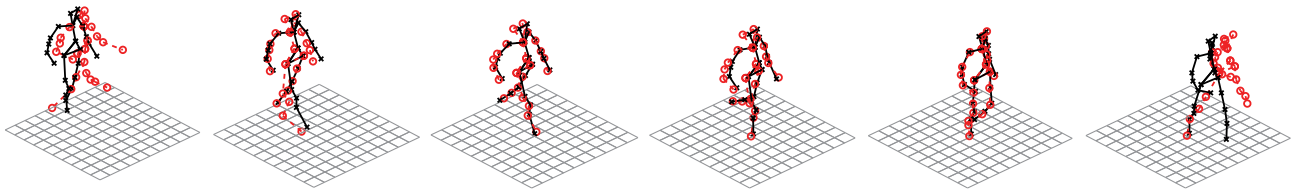
In this section, we evaluate our algorithm quantitatively on motion capture data and qualitatively on real data. In all cases, the trajectory basis vectors are the first K_i DCT basis vectors in order of increasing frequency where K_i is determined by our cross validation scheme. The DCT basis vectors have been shown to provide optimal performance in encoding a signal under the first order Markov process (Hamidi and Pearl 1976) and demonstrated to accurately and compactly model point trajectories (Akhter et al. 2008, 2011). If a 3D trajectory is continuous and smooth, the DCT basis vectors can represent it accurately with relatively few low frequency components. We make the realistic assumption that each point trajectory is continuous and smooth and use the DCT basis as the trajectory basis, $\boldsymbol{\Theta}$. Also for numerical stability, we normalize 2D measurements of the each trajectory such that the mean of 2D measurements is $\mathbf{0}$ and the average distance from the origin is $\sqrt{2}$ before solving Eq. (7) (Hartley 1997). We obtained correspondences of moving points across images, manually. 3D trajectories of moving points are estimated linearly as described in Sect. 3.1. The number of basis vectors is chosen per point using the cross validation method and each linearly estimated trajectory is refined by the nonlinear optimization as described in Sect. 3.2 and in Sect. 3.3, respectively. The results, data, and the code of real data are available on the webpage, http://www.cs.cmu.edu/~hyunsoop/trajectory_reconstruction.html.

5.1 Simulation

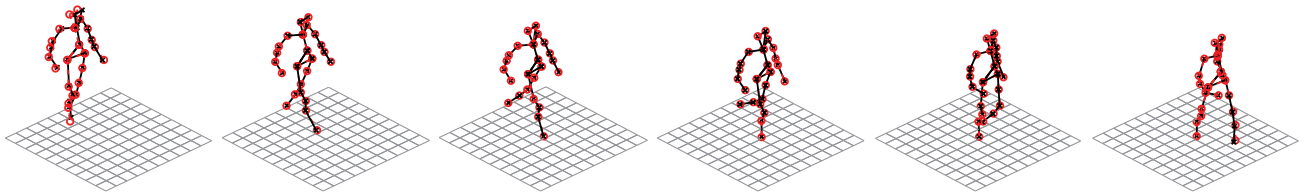
To quantitatively evaluate our method, we generate synthetic 2D images from 3D motion capture data and test it from three perspectives: reconstructability, robustness, and accuracy. For reconstructability, we compare reconstruction by increasing the null component, $e_{\mathbf{C}}$, of the camera trajec-



(a) Trajectory reconstruction at zero reconstructability

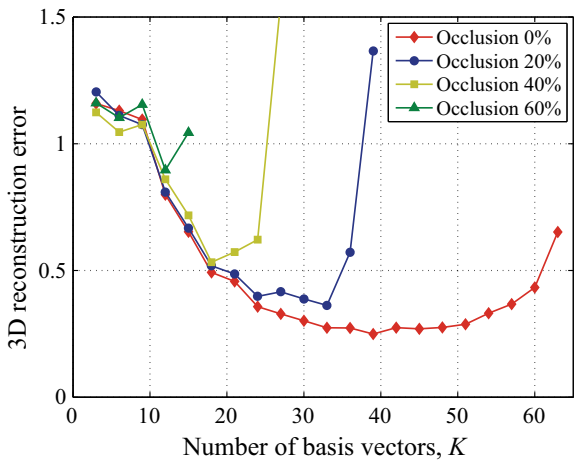


(b) Trajectory reconstruction with low reconstructability

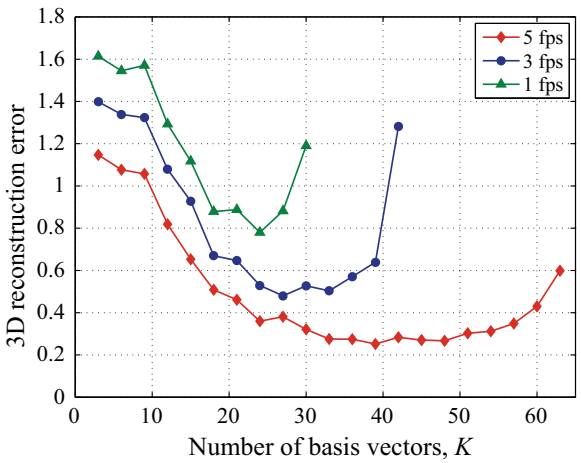


(c) Trajectory reconstruction with high reconstructability

Fig. 9 Qualitative comparison of trajectory reconstruction from various reconstructability. *Black* ground truth, *red* reconstructed trajectory. **a** Zero reconstructability, $\eta = 0$. The relative camera trajectory is stationary and the reconstructed trajectory is exactly the same as the camera trajectory. **b** Low reconstructability, $\eta = 0.32$ results in inaccurate reconstruction at the beginning and the end of the sequence. **c** All trajectories are reconstructed accurately under high reconstructability, $\eta = 5.31$



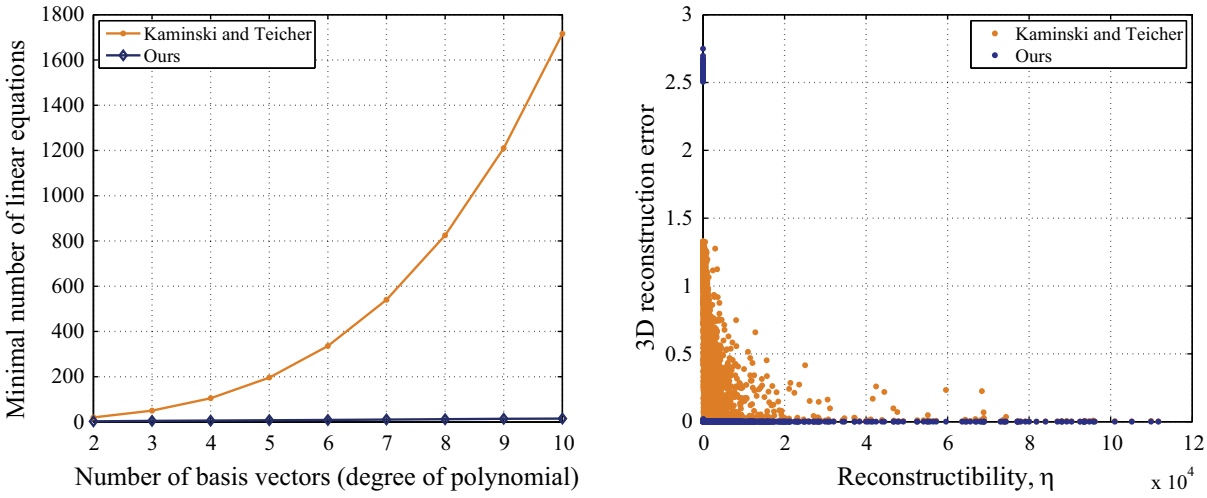
(a) Occlusion



(b) Frame rate

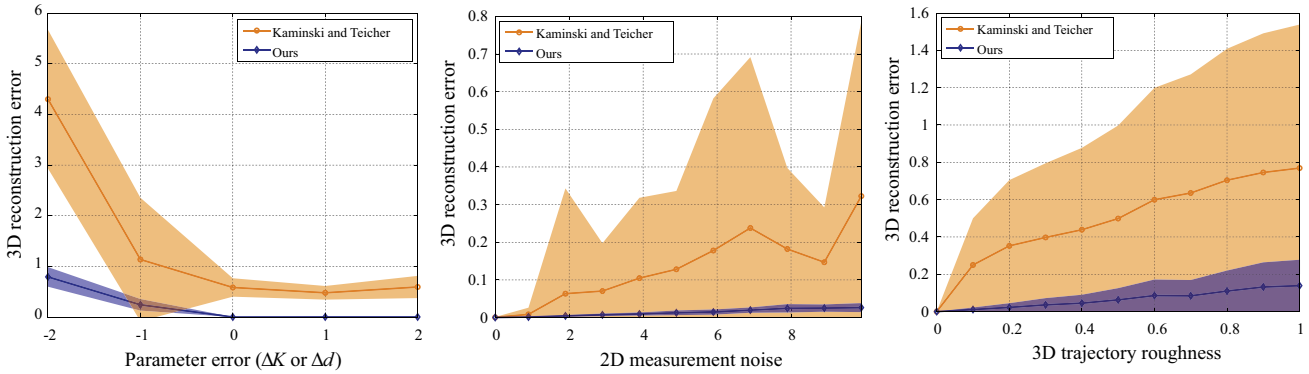
Fig. 10 **a** While a large number of basis vectors results in low 3D reconstruction error in general, reconstruction instability is observed when there is missing data. Reconstruction instability results from overfitting of trajectories. Nevertheless, our algorithm can handle 40 % missing data with 19 basis vectors, which results in relatively low 3D reconstruction error. **b** As frame rate increases, visibility of motion also increases, which results in low 3D reconstruction error

tory. For robustness, we test with missing data and lower sampling rates. Finally, for accuracy, we compare our algorithm with a state-of-the-art algorithm (trajectory triangulation) by Kaminski and Teicher (2004) to the best of our knowledge. The results show our method outperforms their method according to these metrics.



(a) Number of linear equations **(b)** Reconstructability

Fig. 11 **a** The minimal number of linear equations increases exponentially as the degree of polynomial (degree of motion) increases for the method by Kaminski and Teicher (2004) while it increases linearly for our method. This computationally precludes them to reconstruct a trajectory with high complexity. **b** We compare reconstruction accuracy by varying reconstructability. Both methods show inverse relationship between 3D reconstruction error and reconstructability. Our method produces smaller error than their method



(a) Parameter error **(b)** 2D measurement noise **(c)** Trajectory smoothness

Fig. 12 We compare our algorithm with the method proposed by Kaminski and Teicher (2004). We measure reconstruction error as function of error in the input parameters. **a** We show our algorithm can reconstruct a trajectory with high accuracy although the number of basis vectors is mis-estimated while their method cannot. ΔK and Δd are difference between ground truth parameters and estimated parameters. **b** We illustrate the cases where camera poses or 2D projections are inaccurate. **c** We show how much our method can tolerate a trajectory that cannot be modeled by its representation, i.e., non-smooth trajectories. For all cases, our method outperforms their method, i.e., less error and more stable reconstruction. Also, our method exhibits graceful degradation when the error of input parameters increases. Note that the shaded area represents standard deviation of each 3D reconstruction error

5.1.1 Reconstructability

Earlier, we defined the reconstructability of a 3D trajectory as the trade off between the ability of the chosen trajectory basis vectors to accurately describe the point trajectory versus its ability to describe the camera trajectory. To evaluate this effect empirically we relative generate camera trajectories by varying e_C and measure the 3D reconstruction error as shown in Fig. 6. Each trajectory is normalized to have zero mean and unit variance so that errors can be compared across different sequences. Figure 9 shows examples (walking sequences) of trajectory reconstructions under various reconstructability. When the reconstructability is zero shown in Fig. 9a, the reconstructed trajectories are exactly the same as the camera motions because the camera trajectory is the intersection of the hyperplane, l, and the space of trajectory basis vectors, col($\mathbf{\Theta}$), in Fig. 4. When reconstructability is low, $\eta = 0.32$, shown in Fig. 9b, the reconstruction deviates from the ground truth because there is interference from the camera trajectory. High estimation error can be observed at the beginning and

(a) Reprojection error **(b)** Selection per point

Fig. 13 Reprojections of trajectories from manually selected K and automatically selected K_i are shown for the dance scene. **a** *Red cross* measurement, *cyan circle* manually selected K, and *green triangle* automatically and individually selected K_i. Trajectory from K_i has smaller reprojection error. Average reprojections for K and K_i are 11.55 and 6.47, respectively. **b** The number of basis vectors per point is color-coded. The points on the hands require many basis vectors while the points on the left leg which barely move requires few basis vectors

the end of the sequence. If the reconstructability is high, $\eta = 5.31$, reconstruction is very close to the ground truth.

5.1.2 Handling Missing Data

We test for the effects of missing data and low frame rate (sparse measurements) with high reconstructability. Missing samples occur in practice due to occlusion, self-occlusion, or measurement failure.

In general, as the number of the basis vectors K increases, the 3D reconstruction error decreases because the high frequency components of a point trajectory can be described by the basis vectors. However, when there is occlusion, reconstruction instability occurs due to measurement noise. Figure 10a shows the reconstruction error as the amount of occlusion varies (0, 20, 40, and 60 % of the sequence) for different numbers of the DCT basis vectors, K. A walking motion capture sequence was used and each experiment was repeated 10 times with random occlusion. As long as the visibility of a point in a sequence is sufficient to overconstrain Eq. (7), the solution is robust to moderate occlusion. Figure 10a shows that our algorithm can handle relatively high number of missing data (40 %) with $K = 19$.

Figure 10b evaluates the robustness to the frequency of input samples, i.e., varying the effective frame rate of the input sequence given camera motion and point motion. Note that since the camera motion and point motion are fixed, relative motion, or reconstructability, is constant. Visibility of moving points is important to avoid poor conditioning of the solution, and intuitively more frequent visibility results in better reconstruction. The results confirm this observation. As was observed in the occlusion experiment, the higher the truncation factor K, the less the reconstruction error, but reconstruction instability can be observed when frame rate is low (1 fps).

5.1.3 Accuracy

We evaluate our algorithm by comparing with a trajectory recontruction algorithm proposed by Kaminski and Teicher (2004)[7]. The result of this comparison indicates that their method is computationally prohibitive and less fault tolerant.

They represented a trajectory (algebraic curve) as a hypersurface in $\mathbb{P}^5$ where all lines of projections intersect, i.e., a homogeneous polynomial vanishes on Plücker coordinates of 3D lines intersecting the trajectory. The algorithm is composed of two optimizations: to estimate the Chow polynomial from lines of projections and to find points on a trajectory that satisfy the Chow polynomial. To solve the Chow polynomials from lines of projections, $N_d = \binom{d+5}{d} - \binom{d+3}{d-2} - 1$ measurements have to be made, and each measurement produces one linear equation. Therefore, N_d linear equations has to be solved[8] while our method needs to solve $N_K = \lceil 3K/2 \rceil$ linear equations. d is the degree of the homogeneous polynomial that determines degree of motion (complexity of the trajectory), which is equivalent to the number of trajectory basis vectors, K, for our method. N_d increases exponentially while N_K increases linearly as shown in Fig. 11a. This indicates that their method is computationally prohibitive as the degree of motion, d, increases. Inaccurate trajectory reconstruction caused by low reconstructability is also observed from their method as shown in Fig. 11b. 3D reconstruction error is inversely related to reconstructability while their method is more sensitive to reconstructabil-

[7] The method by Avidan and Shashua (2000) can only reconstruct a linear or conic trajectory.

[8] To solve the second part of the optimization, they have to additionally solve $\binom{d+2}{d}$ linear equations.

Table 1 Parameters of real data sequences

	F (s)	# of photos	# of photographers
Rock climbing	39	107	5
Handshake	10	32	3
Speech	24	67	4
Greeting	24	66	4
Dance	16	49	4

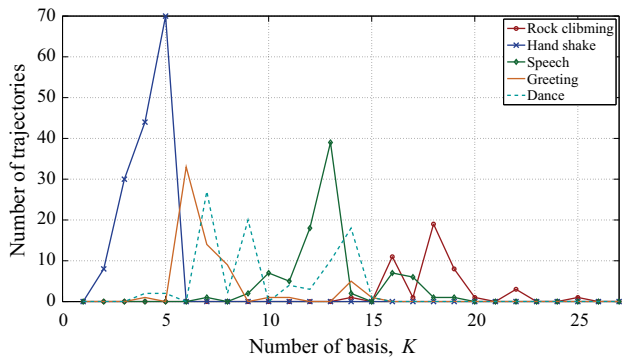


Fig. 14 The distribution of the number of basis vectors. Scenes which are long or contain complex trajectories such as the rock climbing scene or the speech scene (complex hand motions), require the high number of basis vectors while short or simple motion scenes such as the hand shake scene or the greeting scene require the low number of basis vectors. In the greeting scene, there are several trajectories that exhibit a relatively the high number of basis vectors (14–15), which correspond to the hand motion (there is hand waving motion.)

ity than ours, which is shown as a heavy-tailed distribution.

We evaluate both algorithms based on an error tolerance criterion: to what extent can the systems tolerate erroneous input parameters in Fig. 12. Three sources of error are tested: degree of motion error, camera pose estimation error, and a point trajectory model error. Mis-estimated degree of motion, d or K, results in inaccurate reconstruction, i.e., the reconstructed trajectory can be overfitted or oversmoothed. We randomly generate a trajectory with K basis vectors or d degree of polynomial and reconstruct it with K_r and d_r. Figure 12a shows that their algorithm breaks when the trajectory is reconstructed with smaller d_r, i.e., $\Delta d = d_r - d < 0$, while our method does not break significantly for $\Delta K = K_r - K < 0$. When $\Delta d > 0$ and $\Delta K > 0$, the reconstruction is comparable with $d_r = d$ and $K_r = K$. Inaccurate camera pose estimation can produce 2D image measurement noise. We measure 3D reconstruction error as varying Gaussian noise of the projections. Their method easily breaks in the presence of the image noise while our method can reconstruct with high accuracy as shown in Fig. 12b. Finally, we test how much an algorithm can handle a trajectory that cannot be modeled by its representation. Both algorithms model point motion as a smooth trajectory. We generate a 3D smooth trajectory and mix with Gaussian noise to create a non-smooth trajectory. Our algorithm is more tolerant on a non-smooth trajectory with high accuracy than their algorithm, as shown in Fig. 12c. For all cases, our method degrades gracefully as the error of input parameters increases while their method tends to produce erroneous output.

5.2 Experiments with Real Data

The theory of reconstructability states that it is possible to reconstruct 3D point trajectories using the DCT basis vectors if the camera motion relation to the point motion is large. An interesting real world example of this case occurs when many independent photographers take temporally non-coincidental images of the same event from different locations. A collection of these images can be interpreted as the large motion of a camera center that cannot be represented by the DCT trajectory basis vectors. Using multiple photographers, we collected data in several 'media event' scenarios: a person *rock climbing*, a photo-op *hand shake*, a public *speech*, *greeting*, and *dance* (Fig. 13).

The parameters for each scenario are summarized in Table 1. We were able to use the DCT basis vectors for all scenes. The required number of the basis vectors implies the complexity of the trajectory. A long sequence such as the rock climbing scene requires generally the higher number of basis vectors than a short sequence such as hand shake scene as shown in Fig. 14. Figures 15, 16, 17, 18, and 19 show some of input images and reconstructed point trajectories (the number of basis vectors is color-coded into a trajectory).

5.2.1 Camera Pose Estimation

The static scene reconstruction is based on the structure from motion pipeline described in Snavely et al. (2006). Keypoints are extracted by SIFT (Lowe 2004) and all possible pairs of images are considered to find matches using the fundamental matrix. The RANSAC (Fischler and Bolles 1981) based matching enables us to automatically obtain scene correspondences of static points. These correspondences are used to estimate camera poses using structure from motion with incremental bundle adjustment to the image collection. From the first image pair, relative camera pose is estimated from the essential matrix, and then the static correspondences are triangulated. To estimate an additional camera pose we compare the keypoints registered in 3D with new keypoints observed by the target camera and apply a perspective-n-point algorithm (Moreno-Noguer et al. 2007) to estimate the camera pose. If there are unregistered keypoints which are also visible from any of the registered cameras, their 3D locations are estimated through triangulation. This procedure is repeated until no image remains. Camera poses and static structures are also refined by sparse bundle adjustment (Lourakis and

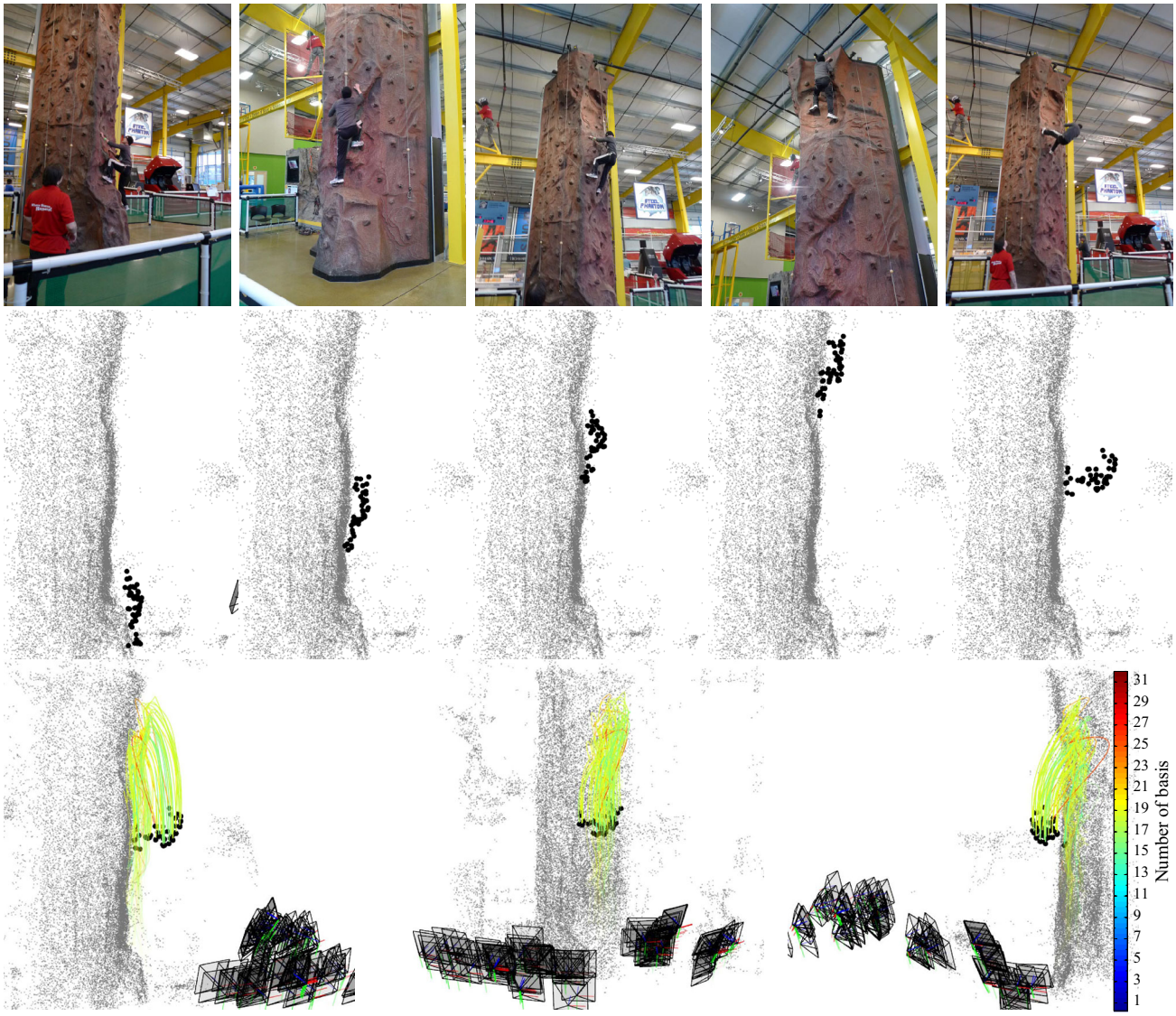


Fig. 15 Results of the rock climbing scene. *Top row* sampled image input, *second row* five snap shots of 3D reconstruction of motion of the rock climber, and *bottom row* reconstructed trajectories in different views. The number of basis vectors is color-coded

Argyros 2009) at each time a new camera is registered. We also extracted time and the focal length of each photo from its EXIF tag.

5.2.2 Selection of the Number of Basis Vectors

To validate the proposed method of selecting the number of basis vectors described in Sect. 3.2, we tested on static points of real scenes where we know $K_i = 1$. As a result, static points of most scenes are classified as $K_i = 1$ (>96 %) except for the speech scene (>70 %). For the speech scene, since the baselines between photographers are very small uncertainty of the depth of points is relatively high. This causes some static points in the speech scene to be classified as moving points in depth direction. Figure 13 shows results of automatic selection of the number of basis vectors for the dance scene. It is compared with $K = 14$ which is set manually for all trajectories. Automatic selection produces smaller reprojection error and it describes point motions better than manual selection.

6 Discussion

We present an algorithm to robustly estimate the general motion of a 3D point from monocular perspective projections. The algorithm is stable in the presence of missing data and measurement error. We characterize the cases when 3D reconstruction is possible and how accurate it can be, based on the relationship between camera motion and point motion.

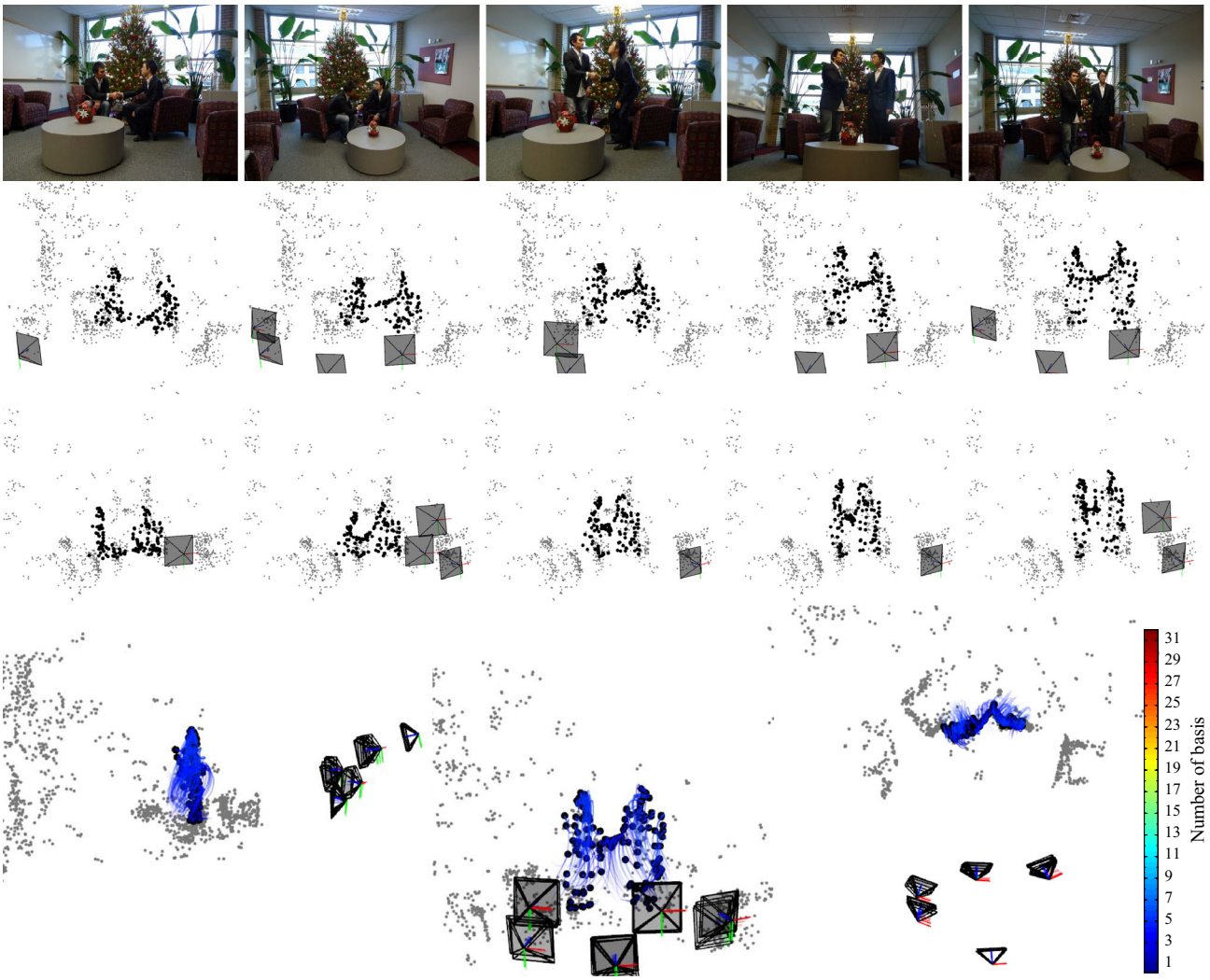


Fig. 16 Results of the handshake scene. *Top row* sampled image input, *second* and *third row* five snap shots of 3D reconstruction in different views, and *bottom row* reconstructed trajectories. The number of basis vectors is color-coded

We also categorize systems as solvable or unsolvable and further define a criterion called reconstructability to characterize the stability of solvable systems. The algorithm presented by Park et al. (2010) is extended to automatically select the number of trajectory basis vectors for each trajectory individually using a cross validation scheme, so as to maximize reconstructability. In addition, we refine the trajectories initialized by the least squares system by minimizing image reprojection error directly.

Our algorithm takes as input the camera pose at each time instant, and pre-defined trajectory basis vectors. These requirements are met in practice when we reconstruct a dynamic scene from collections of images captured by a number of photographers. We estimate the relative camera pose by applying robust structure from motion to the static points in the scene. The DCT is used as pre-defined basis vectors, which we demonstrate is coordinate independent, i.e., remains compact under similarity transforms. Because the effective camera motion in relation to the point motion is sufficiently large, we are able to obtain accurate 3D reconstructions of the dynamic scenes.

Since all points are reconstructed independently, when there are mis-matched correspondences or high depth ambiguity is observed because of small baseline, for instance, the speech scene in Fig. 17, the reconstructed trajectory can be inaccurate. This can be resolved by applying spatial constraints on structure at a given time instant if prior information about 3D structure is available such as a human skeleton model. Future work can explore how spatial constraints may correct trajectories effectively so that the system can reduce the ambiguity of motion.

Our algorithm assumes that the correspondences of moving points are given. We manually specified point correspondences across images for our experiments. From a practical

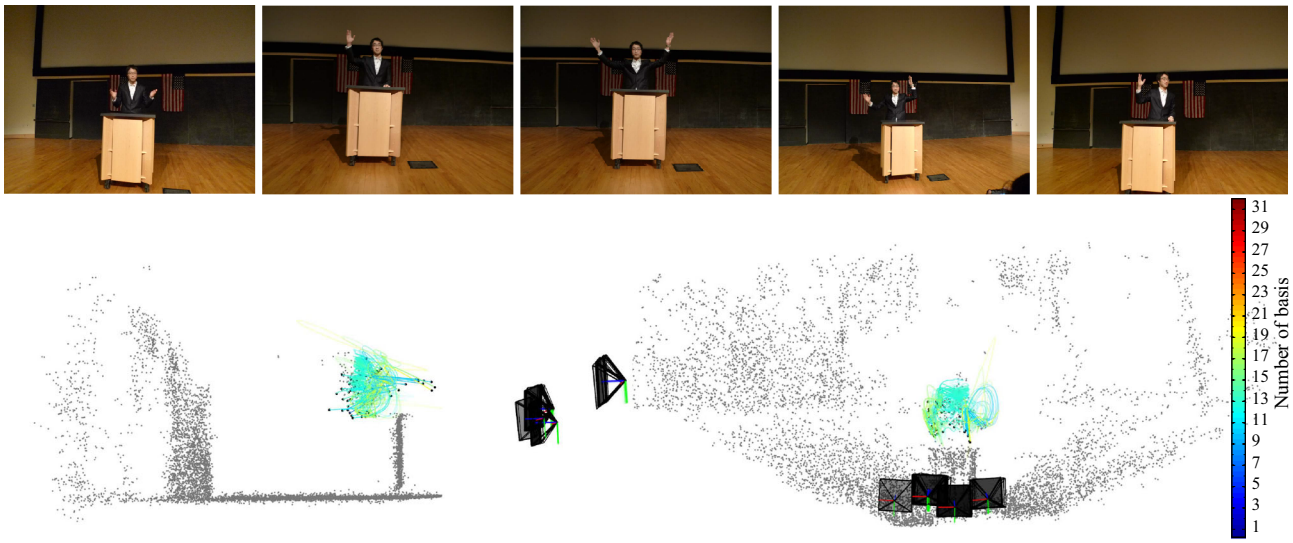


Fig. 17 Results of the speech scene. *Top row* sampled image input, and *bottom row* reconstructed trajectories in different views. The number of basis vectors is color-coded

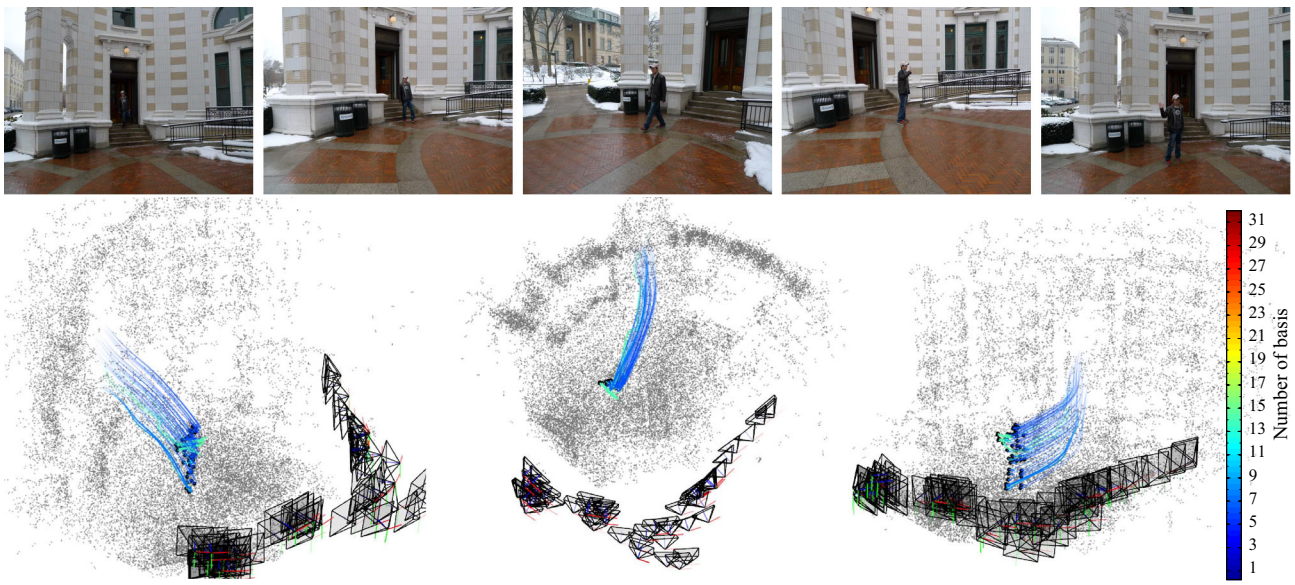


Fig. 18 Results of the greeting scene. *Top row* sampled image input and *bottom row* reconstructed trajectories in different views. The number of basis vectors is color-coded

stand point, this is undesirable. However, as camera optics and sensors improve, and more sophisticated point correspondence methods are developed, the ability to automatically obtain correspondences will likely become achievable. Future directions of this work include making the correspondence process entirely automatic, and applying the method to reconstruct longer sequences where the frequency of photographs, and therefore quality of reconstruction, varies within a sequence. We are also interested in applying stronger priors to recognizable objects like people and faces to construct denser representations.

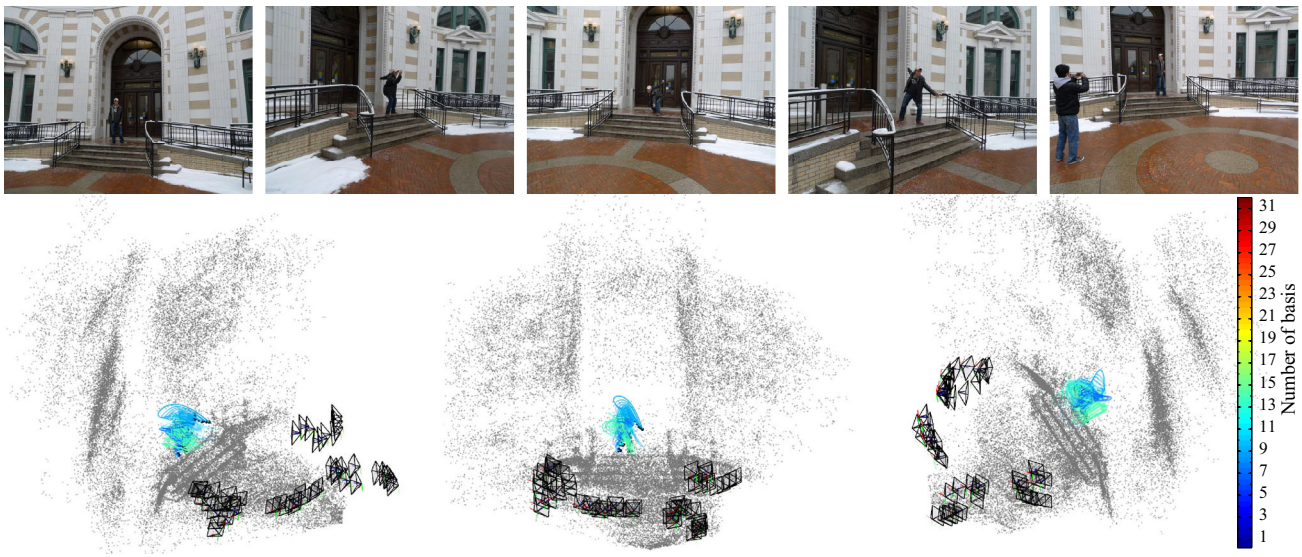


Fig. 19 Results of the dance scene. *Top row* sampled image input, and *bottom row* reconstructed trajectories in different views. The number of basis vectors is color-coded

Acknowledgments This work was supported by NSF Grant IIS-0916272.

Appendix

To prove Result 1, we need to show that the transformed trajectory basis, $\mathbf{S}(\boldsymbol{\Theta})$, span the same space spanned by the original trajectory basis vectors where $\mathbf{S}(\cdot)$ is a similarity transformation, i.e., $\text{col}(\mathbf{S}(\boldsymbol{\Theta})) = \text{col}(\boldsymbol{\Theta})$ where $\text{col}(\boldsymbol{\Theta})$ is a space spanned by the column space of $\boldsymbol{\Theta}$.

Proof (*i*) *scale* $\text{col}(s\boldsymbol{\Theta}) = \text{col}(\boldsymbol{\Theta})$ where s is a scalar.

(*ii*) *translation* translation is spanned by the DC component of $\boldsymbol{\Theta}_{\text{DCT}}$.

(*iii*) *rotation* without loss of generality, the trajectory basis can be rearranged as $\bar{\boldsymbol{\Theta}} = \text{blkdiag}\{\boldsymbol{\theta}, \boldsymbol{\theta}, \boldsymbol{\theta}\}$ where $\boldsymbol{\theta} \in \mathbb{R}^{F \times K}$ is the DCT trajectory basis for each trajectory. The rotated trajectory basis, $(\mathbf{R} \otimes \mathbf{I}_F)\bar{\boldsymbol{\Theta}}$ span the original trajectory basis vectors $\bar{\boldsymbol{\Theta}}$ because,

$$
\begin{aligned}
&\text{col}\left((\mathbf{R} \otimes \mathbf{I}_F)\bar{\boldsymbol{\Theta}}\right) \\
&\quad = \text{col}\left(\begin{bmatrix} R_{11}\mathbf{I}_F & R_{12}\mathbf{I}_F & R_{13}\mathbf{I}_F \\ R_{21}\mathbf{I}_F & R_{22}\mathbf{I}_F & R_{23}\mathbf{I}_F \\ R_{31}\mathbf{I}_F & R_{32}\mathbf{I}_F & R_{33}\mathbf{I}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} & & \\ & \boldsymbol{\theta} & \\ & & \boldsymbol{\theta} \end{bmatrix}\right) \\
&\quad = \text{col}\left(\begin{bmatrix} R_{11}\boldsymbol{\theta} & R_{12}\boldsymbol{\theta} & R_{13}\boldsymbol{\theta} \\ R_{21}\boldsymbol{\theta} & R_{22}\boldsymbol{\theta} & R_{23}\boldsymbol{\theta} \\ R_{31}\boldsymbol{\theta} & R_{32}\boldsymbol{\theta} & R_{33}\boldsymbol{\theta} \end{bmatrix}\right) \\
&\quad = \text{col}\left(\begin{bmatrix} \boldsymbol{\theta} & & \\ & \boldsymbol{\theta} & \\ & & \boldsymbol{\theta} \end{bmatrix} \begin{bmatrix} R_{11}\mathbf{I}_K & R_{12}\mathbf{I}_K & R_{13}\mathbf{I}_K \\ R_{21}\mathbf{I}_K & R_{22}\mathbf{I}_K & R_{23}\mathbf{I}_K \\ R_{31}\mathbf{I}_K & R_{32}\mathbf{I}_K & R_{33}\mathbf{I}_K \end{bmatrix}\right) \\
&\quad = \text{col}\left(\bar{\boldsymbol{\Theta}}(\mathbf{R} \otimes \mathbf{I}_K)\right) \\
&\quad = \text{col}(\bar{\boldsymbol{\Theta}})
\end{aligned}
$$

where $\otimes$ is the Kronecker product, $\mathbf{R}$ is a 3×3 rotation matrix and, $\mathbf{I}_K$ is a $K \times K$ identity matrix. □

References

Akhter, I., Sheikh, Y., & Khan, S. (2009). In defense of orthonormality constraints for nonrigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Akhter, I., Sheikh, Y., Khan, S., & Kanade, T. (2008). Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*.

Akhter, I., Sheikh, Y., Khan, S., & Kanade, T. (2011). Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(7), 1442–1456.

Avidan, S., & Shashua, A. (2000). Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 348–357.

Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S. I., & Sayd, P. (2008). Coarse-to-fine low-rank structure-from-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *ACM transactions on Graphics (SIGGRAPH)*.

Brand, M. (2001). Morphable 3D models from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Brand, M. (2005). A direct method for 3D factorization of nonrigid motion observed in 2D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Bregler, C., Hertzmann, A., & Biermann, H. (1999). Recovering non-rigid 3D shape from image streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dai, Y., Li, H., & He, M. (2012). A simple prior-free method for non-rigid structure-from-motion factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Del Bue, A. (2008). A factorization approach to structure from motion with shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Del Bue, A., Llad, X., & Agapito, L. (2006). Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Faugeras, O., Luong, Q.-T., & Papadopoulou, T. (2001). *The geometry of multiple images: The laws that govern the formation of images of a scene and some of their applications*. Cambridge: MIT Press.

Fayad, J., Agapito, L., & Del Bue, A. (2010). Piecewise quadratic reconstruction of non-rigid surface from monocular sequences. In *Proceedings of the European Conference on Computer Vision*.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Gotardo, P. F. U., & Martinez, A. M. (2011). Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(10), 2051–2065.

Hamidi, M., & Pearl, J. (1976). Comparison of the cosine and Fourier transforms of Markov-I signal. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *24*, 428–429.

Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 580–593.

Hartley, R., & Vidal, R. (2008). Perspective nonrigid shape and motion recovery. In *Proceedings of the European Conference on Computer Vision*.

Kaminski, J. Y., & Teicher, M. (2004). A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, *21*(1), 27–41.

Lladó, X., Del Bue, A., & Agapito, L. (2010). Non-rigid metric reconstruction from perspective cameras. *Image and Vision Computing*, *28*(9), 1339–1353.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, *293*, 133–135.

Lourakis, M. I. A., & Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, *36*(1), 1–30.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. S. (2003). *An invitation to 3-D vision: From images to geometric models*. New York: Springer.

Moreno-Noguer, F., Lepetit, V., & Fua, P. (2007). EPnP: Efficient perspective-n-point camera pose estimation. In *Proceedings of the International Conference on Computer Vision*.

Olsen, S., & Bartoli, A. (2007). Using priors for improving generalization in non-rigid structure-from-motion. In *Proceedings of British Machine Vision Conference*.

Östlund, J., Varol, A., Ngo, D. T., & Fua, P. (2012). Laplacian meshes for monocular 3D shape recovery. In *Proceedings of the European Conference on Computer Vision*.

Ozden, K. E., Cornelis, K., Eychen, L. V., & Gool, L. V. (2004). Reconstructing 3D trajectories of independently moving objects using generic constraints. *Computer Vision and Image Understanding*, *93*, 1453–1471.

Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., & Agapito, L. (2009). Factorization for non-rigid and articulated structure using metric projections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Park, H. S., Shiratori, T., Matthews, I., & Sheikh, Y. (2010). 3D reconstruction of a moving point from a series of 2D projections. In *Proceedings of the European Conference on Computer Vision*.

Salzmann, M., Pilet, J., Ilic, S., & Fua, P. (2007). Surface deformation models for nonrigid 3D shape recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(7), 1481–1487.

Shashua, A., & Wolf, L. (2000). Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *Proceedings of the European Conference on Computer Vision*.

Sidenbladh, H., Black, M. J., & Fleet, D. J. (2000). Stochastic tracking of 3d human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*.

Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (SIGGRAPH)*.

Taylor, J., Jepson, A. D., & Kutulakos, K. N. (2010). Non-rigid structure from locally-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, *9*(2), 137–154.

Torresani, L., Yang, D., Alexander, G., & Bregler, C. (2001). Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Torresani, L., & Bregler, C. (2002). Space-time tracking. In *Proceedings of the European Conference on Computer Vision*.

Torresani, L., Hertzmann, A., & Bregler, C. (2008). Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*, 878–892.

Torresani, L., Hertzmann, A., & Bregler, C. (2003). Learning non-rigid 3D shape from 2D motion. In *Advances in Neural Information Processing Systems*.

Valmadre, J., & Lucey, S. (2012). General trajectory prior for non-rigid reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Vidal, R., & Abretske, D. (2006). Nonrigid shape and motion from multiple perspective views. In *Proceedings of the European Conference on Computer Vision*.

Vidal, R., & Hartley, R. (2004). Motion segmentation with missing data by powerfactorization and generalized pca. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wexler, Y., & Shashua, A. (2000). On the synthesis of dynamic scenes from reference views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wolf, L., & Shashua, A. (2002). On projection matrices $\mathcal{P}^k \rightarrow \mathcal{P}^2, k = 3, \ldots, 6$, and their applications in computer vision. *International Journal of Computer Vision*, *48*(1), 53–67.

Xiao, J., & Kanade, T. (2004). Non-rigid shape and motion recovery: Degenerate deformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, J., Chai, J., & Kanade, T. (2006). A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, *67*(2), 233–246.

Yan, J., & Pollefeys, M. (2005). A factorization-based approach to articulated motion recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhu, S., Zhang, L., & Smith, B. M. (2010). Model evolution: An incremental approach to non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.