Euclidean Structure Recovery from Motion in Perspective Image Sequences via Hankel Rank Minimization

Mustafa Ayazoglu, Mario Sznaier, and Octavia Camps*

Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA

Abstract. In this paper we consider the problem of recovering 3D Euclidean structure from multi-frame point correspondence data in image sequences under perspective projection. Existing approaches rely either only on geometrical constraints reflecting the rigid nature of the object, or exploit temporal information by recasting the problem into a nonlinear filtering form. In contrast, here we introduce a new constraint that implicitly exploits the *temporal ordering* of the frames, leading to a provably correct algorithm to find Euclidean structure (up to a single scaling factor) without the need to alternate between projective depth and motion estimation, estimate the Fundamental matrices or assume a camera motion model. Finally, the proposed approach does not require an accurate calibration of the camera. The accuracy of the algorithm is illustrated using several examples involving both synthetic and real data.

Key words: Structure from Motion, Perspective Images, Rank Minimization

1 Introduction

Recovering 3D structure from a sequence of 2D images has been the subject of substantial research [1, 2]. For the orthographic projection case, Tomasi and Kanade [3] proposed a method based on factorizing a matrix containing the coordinates of the tracked points, which is forced to have at most rank 4. The method has been extended to paraperspective [4, 5] and perspective [6, 7] projection. In the former case, the algorithm relies on the estimation of a set of point–dependent *projective depths*. Sturm and Triggs [6] proposed to recover these depths by using the epipolar constraint between two views, which in turn requires estimating the fundamental matrix. Triggs [7] extended this method by refining the projective depths through an iterative procedure alternating with factorization. Other iterative approaches include [8–11].

Often, factorization techniques are followed by a bundle adjustment to minimize the 2D re-projection error [12–17]. In general, this entails a non-linear optimization based on descend methods which are very sensitive to initialization. [9] avoids this problem by solving a sequence of eigenvalue problems, but convergence cannot be guaranteed.

* This work was supported in part by NSF grants IIS–0713003 and ECCS–0901433, AFOSR grant FA9550–09–1–0253, and the Alert DHS Center of Excellence.

A common feature of the approaches described above is the fact that they rely entirely on geometrical constraints, discarding temporal information[1]. Indeed, most of these methods are based on quasi-linear algorithms that alternate between estimating the structure and projections, and whose convergence cannot be guaranteed [18–20], and the resulting solutions are invariant with respect to frame permutations.

Temporal correlations have been exploited to solve the related problem of simultaneous localization and estimation (SLAM), where the goal is to use data provided by a single moving platform to reconstruct its 3 D trajectory and a local map. In this context, temporal information is exploited by recasting the problem as a non-linear filtering one. The goal is to estimate a state vector that contains the motion state of the moving sensor (e.g. position, velocity, pose) and the 3D coordinates of given features, as well as a probability density function that quantifies the uncertainty in this estimation. Earlier approaches to SLAM required the use of additional sensor data, e.g. odometry or stereo, while later ones, [21] avoid this by requiring a short calibration run using a landmark with a known position. In principle, success of this approach hinges upon the availability of a motion model for the camera, and access to the inputs to the model. While this additional information is typically available in robotic applications, this is not the case for sequences generated by an unknown camera (or object) motion. This difficulty can be circumvented by assuming a simple model (e.g. constant velocity or acceleration [21]), subject to uncertainty. However this leads to larger uncertainty in the estimated feature position. Alternatively, [22] avoid this issue by using the dynamics for tracking only, while reconstructing the 3-D geometry by first triangulating two key-frames obtained during an initialization stage with user input, followed by epipolar search when new keyframes are added and local bundle adjustment. While SLAM methods work well in practice, convergence to the true depths cannot be guaranteed due to uncertainty in the motion model, coupled with the non-convex nature of bundle adjustment. Further, (external) calibration data is usually unavailable in pure SfM applications.

In this paper, we present a convex-optimization based solution to the problem of Euclidean 3D structure recovery from an image sequence under perspective projection. The proposed method avoids the estimation of epipolar geometry and the fundamental matrix. This is accomplished by exploiting the temporal information encoded in the ordering of the given image sequence to recast the problem into a rank minimization form, that can be efficiently solved using existing convex relaxations. The main theoretical result of the paper shows that indeed the solution to this rank-minimization problem recovers the correct Euclidean depths of the scene points, up to a *single* constant scaling factor for all points across the entire motion sequence. This result is general, and neither depends on the object motion model nor necessitates explicitly finding it. The effectiveness of the algorithm is illustrated with several examples involving both synthetic and real data with known ground truth.

[1] In general, the temporal ordering of the frames is *only* used while tracking the features and establishing correspondences across frames.

2 3D Structure from Perspective Images

Consider a camera Cartesian coordinate system defined with its origin at the center of projection and its Z axis along the camera optical axis. Let N be the number of points of a moving rigid object, and let $\mathbf{P}_{ij} = (X_{ij}, Y_{ij}, Z_{ij})^T$ be the 3D Cartesian camera coordinates of point $\mathbf{P}_j$, $j = 1, \ldots, N$, at time i, $i = 1, \ldots, F$. Then, the corresponding 2D image coordinates at time i, $\mathbf{p}_{ij}(u_{ij}, v_{ij})$, are given by

$$u_{ij} = f\frac{X_{ij}}{Z_{ij}} - c_u,\ v_{ij} = \alpha f\frac{Y_{ij}}{Z_{ij}} - c_v \tag{1}$$

where f is the camera's focal length, α is its pixel aspect ratio and (c_u, c_v) is its principal point. In the sequel, for notational simplicity we will assume that $(c_u, c_v) = (0, 0)$[2]. With this notation, the problem of interest here can be formalized as follows.

Problem 1: Given the above setup, find the 3D scene structure $\mathbf{P}_{ij}$ from the $N \times F$ feature correspondences $\mathbf{p}_{ij}$.

Classically, this problem has been solved using the Strum Triggs Algorithm [6], based on iteratively computing the best rank 4 approximation to a matrix constructed from the image data, and the associated projective depths. Since the problem is not jointly convex, this algorithm is guaranteed to converge only to a local solution. Further, the algorithm as stated above can only recover the 3D structure up to an arbitrary (time–varying) projectivity. Recovering the Euclidian geometry entails an additional computationally challenging non–linear, non–convex optimization.

3 Preliminaries:

Below we introduce some preliminary definitions required to recast Problem 1 as a rank minimization problem.

Definition 1. *An operator $\mathcal{L}$ that maps a vector $x_o \in R^n$ to an infinite sequence of vectors $x_k \doteq \{\mathcal{L}[x_o]\}_k \in R^n$ is said to be* point-wise rigid *if*

$$\|\{\mathcal{L}\left[\mathbf{P} - \mathbf{Q}\right]\}_k\|_2 = \|\mathbf{P} - \mathbf{Q}\|_2 \text{ for all } \mathbf{P}, \mathbf{Q}, k$$

Definition 2. *N points $\mathbf{P}_1, \ldots, \mathbf{P}_N \in R^3$ are said to belong to a rigid body if, for each frame k, there exist a point $\mathbf{O}_k \in R^3$ (not necessarily in the object) and a point-wise rigid operator $\mathcal{L}$ such that for all points and all time instants, the corresponding trajectories satisfy: $\mathbf{P}_{ki} - \mathbf{O}_k = \{\mathcal{L}\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]\}_k,\ k = 1, 2, \ldots$ where $\mathbf{P}_{ki}$ denote the coordinates of point $\mathbf{P}_i$ at time k.*

For example, for a constant rotation R about a moving axis we have $\{\mathcal{L}\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]\}_k = R^k\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]$.

Definition 3. *Given a vector sequence $\{\mathbf{y}_k\}_{k=1}^{n+l-1}$ its Hankel matrix is defined as:*

$$\mathbf{H}_{\mathbf{y},n,l} \doteq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_l \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_n & \mathbf{y}_{n+1} & \cdots & \mathbf{y}_{l+n-1} \end{bmatrix}$$

[2] by redefining, if necessary, $\hat{u}_{ij} = u_{ij} + c_u$ and $\hat{v}_{ij} = v_{ij} + c_v$.

4 Recovering Geometry from Hankel Rank Minimization

In this section, we show that the Euclidean structure of a rigid object undergoing a point-wise rigid transformation can be recovered (up to a single scaling factor) by minimizing the rank of the Hankel matrix associated with the trajectory, subject to one linear and two rank constraints. From its definition, it is clear that the rank of the Hankel matrix encapsulates temporal correlations, since it is not invariant under a permutation of the ordering of the frames. The surprising result is that this rank also encapsulates rigidity, since as we prove below, the correct 3D rigid geometry, up to an overall constant scaling factor, is precisely the one that minimizes it, subject to the additional constraints. This result allows for recasting Problem 1 into a rank-minimization form.

Theorem 1 *Consider the image trajectories* $\mathbf{p}_{ki} = (u_{ki}, v_{ki})^T$, $i = 1, 2, 3$, $k = 1, \ldots, F$ *of the perspective projections of three points* $\mathbf{P}_{ki}$, $i = 1, 2, 3$, *belonging to a rigid moving under some point-wise rigid motion operator* $\mathcal{L}$. *Then, the 3D camera Cartesian coordinates of* $\mathbf{P}_{ki}$ $i = 1, 2, 3$, $k = 1, \ldots, F$ *are given by:*

$$\mathbf{P}_{ki} = \begin{bmatrix} X_{ki} \\ Y_{ki} \\ Z_{ki} \end{bmatrix} = \frac{1}{\lambda_o \rho^k} Z^*_{ki} \begin{bmatrix} \frac{1}{f} u_{ki} \\ \frac{1}{\alpha f} v_{ki} \\ 1 \end{bmatrix} \tag{2}$$

where λ_o *and* $\rho > 0$ *are constant factors (point and frame independent), and where* $\{Z^*_{k1}, Z^*_{k2}, Z^*_{k3}\}_{k=1,\ldots,F}$ *solve the following rank minimization problem*

$$\min_{\{Z^*_{k1}, Z^*_{k2}, Z^*_{k3}\}_{k=1,\ldots,F}} \operatorname{rank}\left(\begin{bmatrix} \mathbf{H}_{\mathbf{y}^{13}} & \mathbf{H}_{\mathbf{y}^{23}} \end{bmatrix}\right) \text{ subject to: } Z_{ki} \geq 1 \tag{3}$$

where

$$\mathbf{y}^{ij}_k = \begin{bmatrix} \frac{1}{f}(Z^*_{ki} u_{ki} - Z^*_{kj} u_{kj}) \\ \frac{1}{\alpha f}(Z^*_{ki} v_{ki} - Z^*_{kj} v_{kj}) \\ Z^*_{ki} - Z^*_{kj} \end{bmatrix}$$

and $\mathbf{H}_{\mathbf{y}} \doteq \mathbf{H}_{\mathbf{y}, \lfloor F/2 \rfloor, F}$, *the Hankel matrix of the sequence* $\{\mathbf{y_k}\}_{k=1}^F$.

Proof: See the Appendix.

Theorem 1 allows for recovering the correct *relative* 3D structure by solving a rank–minimization problem. This follows from the fact that since $Z^*_{ki} = \lambda_o \rho^k Z_{ki}$, then $\frac{Z^*_{ki}}{Z_{ki}} = \frac{Z^*_{kj}}{Z_{kj}}$ for all (i, j), where Z and Z^* denote the actual and recovered depths, respectively. While in many situations this may suffice, in others it is of interest to recover the geometry up to an overall, frame-independent scaling. As we show next, this can be accomplished by adding one linear and two rank constraints to the problem.

Corollary 1 *The correct 3D geometry (up to a single constant scaling factor) satisfies* (3)*, subject to one linear and two rank constraints.*

Proof. Note that the solutions to (3) satisfy: $\|\mathbf{P}_{ki} - \mathbf{P}_{kj}\|_2^2 = \left(\frac{1}{\lambda_o \rho^k}\right)^2 \|\mathbf{P}^*_{ki} - \mathbf{P}^*_{kj}\|_2^2$ where $\mathbf{P}^*_{ki} \doteq Z^*_{ki} \left[\frac{u_{ki}}{f} \;\; \frac{v_{ki}}{\alpha f} \;\; 1\right]^T$. Next, impose rigidity of the reconstructed trajectories

only across the first and last frames, leading to:

$$\begin{aligned} 0 &= \|\mathbf{P}^*_{Fi} - \mathbf{P}^*_{Fj}\|_2^2 - \|\mathbf{P}^*_{1i} - \mathbf{P}^*_{1j}\|_2^2 \Rightarrow \\ 0 &= \left(\lambda_o \rho^F\right)^2 \|\mathbf{P}_{Fi} - \mathbf{P}_{Fj}\|_2^2 - \left(\lambda_o \rho\right)^2 \|\mathbf{P}_{1i} - \mathbf{P}_{1j}\|_2^2 \Rightarrow \rho = 1 \end{aligned} \tag{4}$$

where the last equality follows from the fact that the actual trajectories satisfy $\|\mathbf{P}_{ki} - \mathbf{P}_{kj}\|_2 = \text{constant}$, for all k. Thus, imposing rigidity of the reconstructed object *only* for 2 points across the first and last frames forces the overall scaling to become frame independent (e.g. $\alpha_k = \lambda_o(1)^k = \lambda_o$). As we show below, the constraint (4) can be recast as a combination of linear and rank constraints. Start by rewriting the constraint $\||\mathbf{P}^*_{11} - \mathbf{P}^*_{12}\|_2^2 = \||\mathbf{P}^*_{F1} - \mathbf{P}^*_{F2}\|_2^2$ as:

$$\begin{aligned} &Z_{11}^2\left(\frac{u_{11}^2}{f^2} + \frac{v_{11}^2}{f^2\alpha^2} + 1\right) + Z_{12}^2\left(\frac{u_{12}^2}{f^2} + \frac{v_{12}^2}{f^2\alpha^2} + 1\right) - 2 * Z_{11}Z_{22}\left(\frac{u_{11}u_{12}}{f^2} + \frac{v_{11}v_{12}}{f^2\alpha^2} + 1\right) - \\ &Z_{F1}^2\left(\frac{u_{F1}^2}{f^2} + \frac{v_{F1}^2}{f^2\alpha^2} + 1\right) - Z_{F2}^2\left(\frac{u_{F2}^2}{f^2} + \frac{v_{F2}^2}{f^2\alpha^2} + 1\right) + 2 * Z_{F1}Z_{F2}\left(\frac{u_{F1}u_{F2}}{f^2} + \frac{v_{F1}v_{F2}}{f^2\alpha^2} + 1\right) = 0 \end{aligned} \tag{5}$$

Next, define the following variables:

$$m_t^{20} \doteq Z_{t1}^2,\ m_t^{11} \doteq Z_{t1}Z_{t2},\ m_t^{02} \doteq Z_{t2}^2 \tag{6}$$

In terms of these new variables, (5) can be rewritten as the *linear* constraint:

$$\begin{aligned} &m_1^{20}\left(\frac{u_{11}^2}{f^2} + \frac{v_{11}^2}{f^2\alpha^2} + 1\right) + m_1^{02}\left(\frac{u_{12}^2}{f^2} + \frac{v_{12}^2}{f^2\alpha^2} + 1\right) - 2 * m_1^{11}\left(\frac{u_{11}u_{12}}{f^2} + \frac{v_{11}v_{12}}{f^2\alpha^2} + 1\right) - \\ &m_F^{20}\left(\frac{u_{F1}^2}{f^2} + \frac{v_{F1}^2}{f^2\alpha^2} + 1\right) - m_F^{02}\left(\frac{u_{F2}^2}{f^2} + \frac{v_{F2}^2}{f^2\alpha^2} + 1\right) + 2 * m_F^{11}\left(\frac{u_{F1}u_{F2}}{f^2} + \frac{v_{F1}v_{F2}}{f^2\alpha^2} + 1\right) = 0 \end{aligned} \tag{7}$$

Further, it can be easily seen[3] that (6) is equivalent to

$$\text{rank}\left\{\begin{bmatrix} m_t^{20} & m_t^{11} \\ m_t^{11} & m_t^{02} \end{bmatrix}\right\} = 1,\ t = \{1, F\} \tag{8}$$

□

From this corollary, it follows that the 3D geometry (up to a single scaling factor) of a moving rigid object can be found by using the following algorithm.

Algorithm 1: RANK MINIMIZATION BASED 3D-DEPTH RECOVERY

Data: Camera Intrinsic Parameters.
Input: (u_{ki}, v_{ki}), the temporally ordered 2-D coordinates of N points in F frames.
Output: 3D depths Z_{ki} up to an overall scaling constant.
1. Form the *difference* vectors $\mathbf{y}_k^{iN} \doteq \mathbf{P}^*_{ki} - \mathbf{P}^*_{kN},\ i = 1, \ldots, N-1$ where $\mathbf{P}^*_{ki} \doteq Z^*_{ki}\left[\frac{u_{ki}}{f}\ \frac{v_{ki}}{\alpha f}\ 1\right]^T$, and the corresponding Hankel matrices $\mathbf{H}_{\mathbf{y}^{iN}}$
2. Solve: $\min_{Z^*_{ki} \geq 1} \text{rank}\left[\mathbf{H}_{\mathbf{y}^{1N}} \ldots \mathbf{H}_{\mathbf{y}^{N-1N}}\right]$ subject to (7) and (8)

[3] This follows from simply decomposing the matrix as $\mathbf{M} = \mathbf{v}^T\mathbf{v}$, with $\mathbf{v}^T = \begin{bmatrix} Z_{t1} & Z_{t2} \end{bmatrix}$.

4.1 Computational Complexity and Robustness Considerations.

In principle, Algorithm 1 will recover the unknown Z_{ij} in a single optimization step. Moreover, although rank minimization is generically NP–hard, efficient convex relaxations are available. In particular, in this paper we used the LMIRank relaxation [23]. A potential problem here is the computational cost entailed in solving simultaneously for all Z_{ki}, since the computational complexity of this relaxation scales as (number of decision variables)^5. On the other hand, using larger sets of points minimizes the effects of outliers. To balance these effects we pursued a RANSAC (Random Sample Consensus) [24] approach. Since the minimum number of points required to define a 3D coordinate system is 4, we proceeded by finding the 3D coordinates corresponding to 4 points, randomly selected from the complete set of image points, N_s times. Out of these 4-tuples, the one preserving rigidity the most was used to find the coordinates of the remaining points by exploiting the fact that the measurements matrix has at most rank 4. Thus, given the 3D trajectories of 4 points $\mathbf{P}_{ki}$, the depth of a fifth point Z_{k5} can be found by solving a problem of the form: $\min_{\mathbf{s},Z_{k5}} \|\mathbf{W}\cdot\mathbf{s}-\mathbf{P}_5\|$, where

$$\mathbf{W}=\begin{bmatrix}\mathbf{P}_{11} & \dots & \mathbf{P}_{14}\\ \vdots & \dots & \vdots\\ \mathbf{P}_{F1} & \dots & \mathbf{P}_{F4}\end{bmatrix};\ \mathbf{P}_5 \doteq \left[\tfrac{1}{f}Z_{15}u_{15}\ \tfrac{1}{f\alpha}Z_{15}v_{15}\ Z_{15}\ \dots\ \tfrac{1}{f}Z_{F5}u_{F5}\ \tfrac{1}{f\alpha}Z_{F5}v_{F5}\ Z_{F5}\right]^T$$

5 Experiments

The accuracy of the proposed algorithm is illustrated next with experiments using synthetic and real data. In all cases, the 3D structure recovered using our algorithm (HankelSFM), is compared against the results of the Hung and Tang (HTSFM) and Mahamud and Hebert (MHSFM) algorithms. Videos of the data are provided as additional material.

5.1 Synthetic Data: the Utah Teapot

Next, we illustrate the robustness of the proposed algorithm to noise and poor calibration data. The data consists of the trajectories of the perspective projections of 137 points[4] on the Utah Teapot, centered at $(880, 250, 860)'$, as seen by a pin-hole camera with focal length $f = 400$ and image size 800×600 pixels.

In the first experiment, the teapot underwent a constant angular velocity rotation $\omega_r = 0.3$, around the axis $a = (0, 0, 1)'$, while in the second experiment, the camera is also translated with constant velocity $(-10, 5, 0)'$. Figure 1 (a)–(d) shows renderings for frames 1, 5 and 10 for the rotation experiment and the corresponding reconstructions using HankelSFM, HTSFM and MHSFM. As shown there, HankelSFM preserves the Euclidean geometry while the other methods deform the object frame to frame. Quantitative comparisons are given in Figures 1 (e)-(f), 2 and 3. Figure 1 (e)-(f) shows

[4] Nine points were selected from each surface of the Teapot.

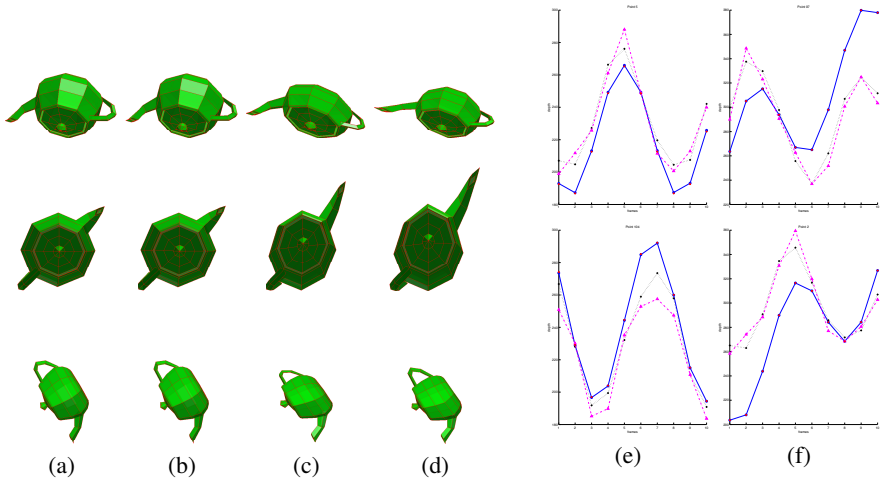


Fig. 1. (a): Frames 1, 5, and 10 of the actual teapot sequence. (b)–(d): 3D structure recovered using HankelSFM (b), MHSFM (c) and HTSFM (d). Note that HankelSFM does not introduce geometric distortion between frames. Right:Real and estimated depth trajectories for two basis points. Red stars: ground truth data; solid blue line: HankelSFM; dotted black line: MHSFM; and dashed magenta line: HTSFM. (e) Rotation experiment. (f) Rotation and translation experiment.

the depth trajectories of two of the four points selected as basis points by the HankelSFM method, and the depths recovered using the three algorithms. All trajectories were scaled by the *single* scaling factor $c = \sum_k \sum_i Z_{ki} / \sum_k \sum_i Z^*_{ki}$ where Z_{ki} and Z^*_{ki} are the ground truth and the estimated depth for point i at frame k, respectively. Since the data is noiseless, HankelSFM exactly recovers the geometry (up to the scaling factor c) as expected, while the other methods introduce varying distortion across frames. Quantitatively, the distortion for all the points can be seen in Figure 2, showing the plots of the differences between the ratio of the elements of W and W^*, the true and reconstructed 3D measurement matrices, respectively, and the normalization factor c . As shown there, only the HankelSFM method produces a flat surface indi-

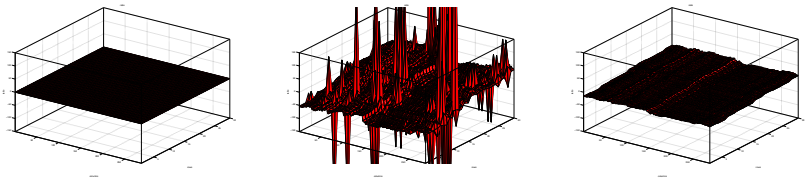


Fig. 2. $\frac{W}{W^*} - c$ for the translation and rotation Utah Teapot experiment. (a) HankelSFM. (b) MHSFM. (c) HTSFM.

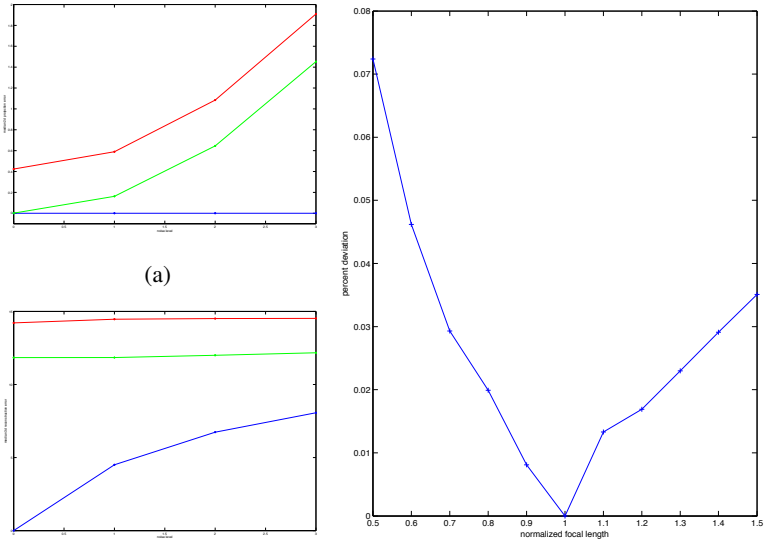


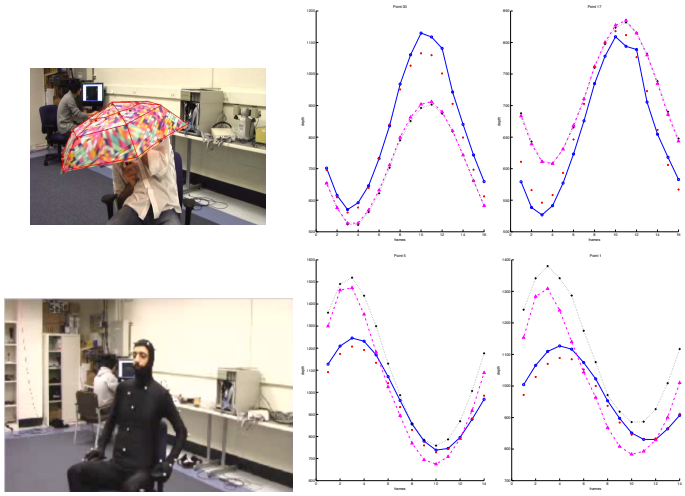
Fig. 3. (a) 2D re-projection and (b) 3D reconstruction median error as noise is increased from 0 to 3 pixels (blue HankelSFM, red MHSFM and green HTSFM). (c) Scaling factor variation Δ as the focal length used by the algorithm is varied from 0.5 to 1.5 times the true focal length.

cating a uniform scaling factor across all frames. Additionally, table 1 summarizes the 3D and the 2D re-projection median error for the three methods (noiseless data) while Figures 3 (a) and (b) plot them for increasing noise levels up to 3 pixels. In all cases, the errors are significantly lower for HankelSFM than for MHSFM and HTSFM. Finally, the very small effect of the choice of focal length on the accuracy of the depth estimation is illustrated in Figure 3 (c) where the relative variation of the scaling factor $\Delta = \max_{k,i} \|Z_{ki}/Z^*_{ki} - c\|/c$ is plotted against K, as the focal length used by the algorithm is set to Kf where f is the true focal length and $0.5 \leq K \leq 1.5$.

5.2 Real Data with Ground Truth

The purpose of these experiments is to compare the performance of HankelSFM against HTSFM and MHSFM using real data. In order to asses the accuracy of the algorithms, the 2D data was generated by projecting the *noisy* 3D coordinates of special markers attached to an umbrella and to a human sitting on a swivel chair that were measured using a VICON motion capture system[5] as shown in Figure 4, left. Quantitative results and comparisons between the 3D reconstructions and ground truth are displayed

[5] It should be noted that the objects used in these experiments are flexible. Furthermore, the markers are about 1cm. in diameter and hence have a significant depth which affects the measurement of their location as the object moves in front of the motion capturing system.



(a) (b)

Fig. 4. (a) Sample frames of the Umbrella (top) and Human on a chair (bottom) sequences. (b) Estimated depth trajectories for two basis points. Red stars: ground truth data; solid blue line: HankelSFM; dotted black line: MHSFM; and dashed magenta line: HTSFM.

in Figures 4, right, and 5. Finally, 3D and 2D re-projection errors are summarized in Table 1. As shown there, the HankelSFM algorithm recovers 3D structure up to a *unique* constant and its 3D accuracy outperforms the other two algorithms.

Table 1. 3D and 2D re-projection median error.

Data Set	HankelSFM		MHSFM		HTSFM	
	3D (mm.)	2D (pixels^2)	3D (mm.)	2D (pixels^2)	3D (mm.)	2D (pixels^2)
Teapot (R)	4.89e-1	0	1.34e+1	3.5e+0	1.34e+1	1.2e-7
Teapot(RT)	1.61e-4	0	3.00e+1	1.0e+0	3.20e+1	2.5e-7
Umbrella	3.50e+1	0	8.22e+1	0.6176	8.32e+1	0.0136
Human	4.10e+1	0	1.37e+2	2.3091	1.51e+2	0.2713

6 Conclusions

In this paper we propose a novel algorithm for 3D Euclidean structure recovery from image sequences under perspective projection. The main idea is to exploit geometrical information encapsulated in the rank of a matrix (the Hankel matrix) constructed from the measurements. This rank implicitly encapsulates temporal information, since it strongly depends on the temporal order of the sequence: the Hankel matrices corresponding to two sequences with the same data in different order have generically different rank. The main result of the paper shows that the provably correct depths (up

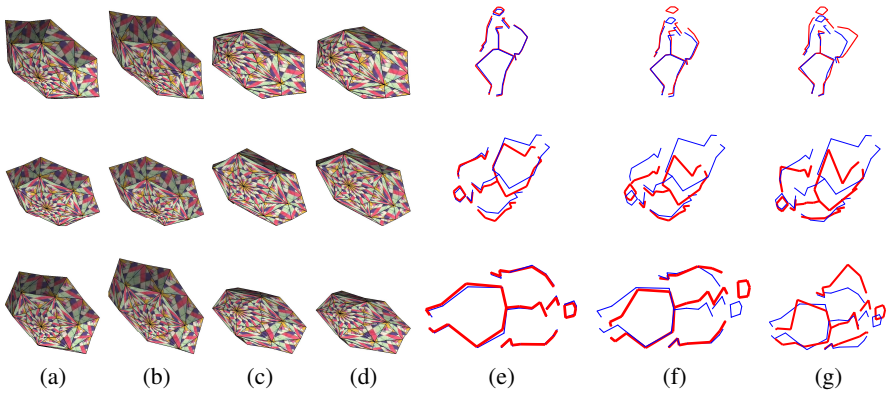


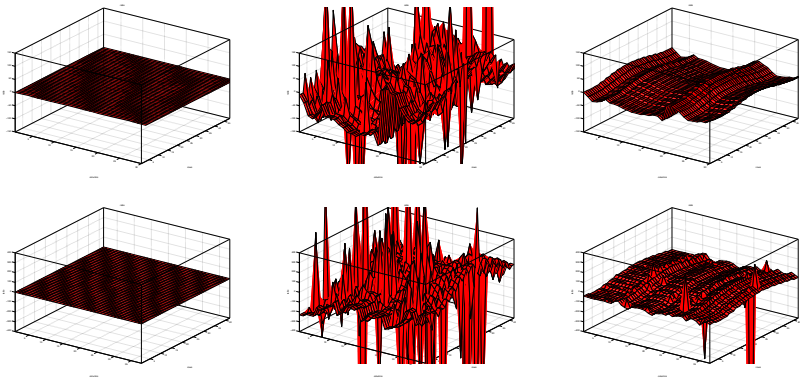
Fig. 5. Left: Frames 1, 6 and 12 of the umbrella sequence. (a) Ground truth data, and 3D structure recovered using (b) HankelSFM, (c) MHSFM and (d) HTSFM. Right: Frames 1, 7, 14 frames of the human on a chair sequence with ground truth data (blue) superimposed with 3D structure (red) recovered using (e) HankelSFM, (f) MHSFM and (g) HTSFM.

to an arbitrary, overall scaling constant) are the ones that minimize the rank of the corresponding Hankel matrix, thus allowing for recasting the SfM problem into a rank minimization one. This result was established by exploiting the existence of an underlying model governing the motion of the rigid body. However, no assumptions are made about this model, and there is no need to find its parameters. Indeed, our results hold independently of the object motion model. While rank-minimization problems are NP hard, recent developments in the field allow for relaxing them to a convex optimization form that can be efficiently solved. When compared to existing approaches, the proposed algorithm recovers the 3D geometry, up to a single arbitrary scaling constant, and does require neither solving a challenging non-linear optimization, performing bundle adjustment, external camera calibration or the availability of a motion model for the moving object.

The advantages of the proposed algorithm were illustrated with synthetic and real image sequences. Research is currently underway seeking to extend these results to articulated and non-rigid objects.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge Univeristy Press (2003)
2. Faugeras, O.D., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press (2001)
3. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992) 137–154
4. Morita, T., Kanade, T.: A paraperspective factorization method for recovering shape and motion from image sequences. IEEE Trans. on PAMI **19** (1997) 858–867



(a) (b) (c)

Fig. 6. $\frac{W}{W^*} - c$ for the umbrella (top row) and for the human on a chair (bottom row) sequences. (a) HankelSFM. (b) MHSFM. (c) HTSFM.

5. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE Transactions on PAMI **19** (1997) 206–218
6. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In Buxton, B., Cipolla, R., eds.: Proceedings of the 4th ECCV, Cambridge, England. Volume 1065 of LNCS., Springer-Verlag (1996) 709–720
7. Triggs, B.: Factorization methods for projective structure and motion. In: IEEE CVPR. (1996)
8. Sparr, G.: Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In: Int. Conf. on Pattern Recognition. (1996)
9. Chen, G., Medioni, G.: Efficient iterative solutions to m-view projective reconstruction problem. In: IEEE CVPR. Volume 2. (1999) 55–61
10. Mahamud, S., Hebert, M.: Iterative projective reconstruction from multiple views. In: IEEE CVPR. Volume 2. (2000) 430–437
11. Hung, Y., Tang, W.: Projective reconstruction from multiple views with minimization of 2d reprojection error. International Journal of Computer Vision **66** (2006) 305–317
12. Mohr, R., Veillon, F., Quan, L.: Relative 3d reconstruction using multiple uncalibrated images. In: IEEE CVPR. (1993) 543–548
13. Hartley, R.: Euclidean reconstruction from uncalibrated views. In Mundy, J., Zisserman, A., eds.: Applications of Invariance in Computer Vision. Volume LNCS 825. (1993) 237–256
14. Morris, D., Kanatani, K., Kanade, T.: Euclidean reconstruction from uncalibrated views. In: Vision Algorithms Theory and Practice. Volume Springer LNCS. (1999)
15. Shum, H.Y.and Ke, Q., Zhang, Z.: Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In: IEEE CVPR. (1999)
16. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment-a modern synthesis. In Triggs, W., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and Practice. Volume LNCS 1883. Springer Verlag (2000) 298–375
17. Bartoli, A., Sturm, P.: Three new algorithms for projective bundle adjustment with minimum parameters. Technical Report 4236, INRIA (2001)
18. Oliensis, J.: Fast and accurate self-calibration. In: ICCV. (1996) 745–752

19. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably convergent iterative methods for projective structure frommotion. In: IEEE CVPR. (2001) 1018–1025
20. Oliensis, J., Hartley, R.: Iterative extensions of the strum/triggs algorithm: convergence and nonconvergence. IEEE Trans. on PAMI **29** (2007) 2217–2233
21. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real time single camera slam. IEEE Trans. on PAMI **29** (2007) 1052–1067
22. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. ISMAR'07, Nara, Japan (2007)
23. Orsi, R.: LMIRank: software for rank constrained lmi problems. (web page and software) (2005) `http://rsise.anu.edu.au/robert/lmirank/`.
24. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm. ACM **24** (1981) 381–395
25. Kailath, T.: Linear Systems. Prentice Hall (1980)

A Proof of Theorem 1.

The proof, based on basic concepts from Linear Systems theory (see for instance the textbook [25]), consists of three steps:

1. Find an operator $\mathcal{L}$ with 2 inputs, such that its response to an impulse applied at the i^{th} input is precisely $\mathbf{y}_k^{i,\alpha_{ki}} \doteq (\alpha_{ki}\mathbf{P}_{ki} - \alpha_{k3}\mathbf{P}_{k3})$.
2. Use a realization of $\mathcal{L}$ to find the minimal rank of any linear time varying operator that interpolates the data, and to establish that the minimum rank interpolant is time-invariant and corresponds to the case $\alpha_{ki} = \lambda_o \rho^k$, for some $\lambda_o, \rho > 0$.
3. Use the connection between rank of a Linear Time Invariant (LTI) operator and the rank of its associated Hankel matrix to establish that minimizing the rank of $\mathbf{H}_{\mathbf{y}_{ki}^{\alpha}}$ recovers the depths Z_{ti} up to an overall scaling factor of the form $\alpha_t = \lambda_o \rho^t$.

Step 1; Assume[6], that the Markov parameters of $\mathcal{L}$ and $\mathbf{O}_k$ satisfy:

$$\mathbf{L}_t = \textstyle\sum_{i=1}^{n_L} \mathbf{A}_i^L \mathbf{L}_{t-i},\ \mathbf{O}_t = \sum_{i=1}^{n_O} \mathbf{A}_i^O \mathbf{O}_{t-i}, \qquad \mathbf{A}_i^L, \mathbf{A}_i^O \in R^{3\times 3} \tag{9}$$

Let $\mathbf{x}_t^i \doteq \mathbf{P}_{ti} - \mathbf{O}_t$. From the above, it follows that the trajectories $\mathbf{x}_k^i$ also satisfy a model of the form

$$\mathbf{x}_t^i = \sum_{j=1}^{n_L} \mathbf{A}_j^L \mathbf{x}_{t-j}^i, \tag{10}$$

or, in compact form:

$$\begin{aligned} \xi_{t+1}^i &= \mathcal{A}_L \xi_t^i, \\ \mathbf{x}_t^i &= \mathcal{C}_L \xi_t^i \end{aligned} \tag{11}$$

where

$$\mathcal{A}_L \doteq \begin{bmatrix} \mathbf{A}_1^L & \mathbf{A}_2^L & \dots & \mathbf{A}_{n_L-1}^L & \mathbf{A}_{n_L}^L \\ \mathbf{I} & 0 & \dots & \dots & 0 \\ 0 & \mathbf{I} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{I} & 0 \end{bmatrix} \quad \xi_t^i \doteq \begin{bmatrix} \mathbf{x}_{t-1}^i \\ \mathbf{x}_{t-2}^i \\ \vdots \\ \mathbf{x}_{t-n_L}^i \end{bmatrix}, \ \mathcal{C}_L = \begin{bmatrix} \mathbf{I}\, \mathbf{0} \dots \mathbf{0} \end{bmatrix}$$

[6] This is without loss of generality, since over finite horizons, any trajectory $\mathbf{L}_k$ can be interpolated with an ARMA model of sufficiently high order.

With this notation, the trajectories $\mathbf{x}_t^i$ in (10) are given by:

$$\mathbf{x}_t^i = \mathcal{C}_L \xi_t^i = \mathcal{C}_L \mathcal{A}_L \xi_{t-1}^i = \cdots = \mathcal{C}_L \mathcal{A}_L^t \xi_o^i \tag{12}$$

Thus, $\mathbf{x}_t^i$ is the impulse response of the system:

$$\begin{aligned} \xi_{t+1}^i &= \mathcal{A}_L \xi_t^i + \xi_o^i \delta_t \\ \mathbf{x}_t^i &= \mathcal{C}_L \xi_t^i \end{aligned} \tag{13}$$

A similar situation holds for $\mathbf{O}_t$, with $\mathbf{A}_L^j$ and $\mathcal{A}_L$ replaced by $\mathbf{A}_O^j$ and $\mathcal{A}_O$, respectively, and ξ_t by a vector ω_t containing the past values $\mathbf{O}_k,\ k = t, \ldots, t - n_O + 1$. Hence $\mathbf{O}_t$ can be obtained as the impulse response of a system with state space realization $(\mathcal{A}_O, \omega_o, [\mathbf{I}\ \mathbf{0}\ \ldots\ \mathbf{0}])$.

Given two points $\mathbf{P}_i, \mathbf{P}_j$ from the rigid, and a time varying scaling constant α_t, consider now the vector $\mathbf{y}_t^{\alpha_t} \doteq (\alpha_t \mathbf{P}_{ti} - \mathbf{P}_{tj})$. Since $\mathbf{P}_{ti} = \mathbf{x}_t^i + \mathbf{O}_t$, we have that

$$\mathbf{y}_t^{\alpha_t} = \alpha_t(\mathbf{x}_t^i + \mathbf{O}_t) - (\mathbf{x}_t^j + \mathbf{O}_t) = \alpha_t \mathbf{x}_t^i - \mathbf{x}_t^j + (\alpha_t - 1)\mathbf{O}_t$$

From (13) and linearity it follows that the trajectory $\mathbf{y}_t^{\alpha_t}$ can be generated as the impulse response of the system:

$$\begin{aligned} \zeta_{t+1} &= \begin{bmatrix} \mathcal{A}_L & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_L & \mathbf{0} \\ \mathbf{O} & \mathbf{O} & \mathcal{A}_O \end{bmatrix} \zeta_t + \begin{bmatrix} \xi_o^i \\ \xi_o^j \\ \omega_o \end{bmatrix} \delta_t \\ \mathbf{y}_t^{\alpha_t} &= \begin{bmatrix} \alpha_t \mathcal{C}_L & -\mathcal{C}_L & (\alpha_t - 1)\mathcal{C}_O \end{bmatrix} \zeta_t \end{aligned} \tag{14}$$

Finally, consider three points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and the corresponding vectors $\mathbf{y}^{i\alpha_{ti}} \doteq \alpha_{ti}\mathbf{P}_{ti} - \alpha_{t3}\mathbf{P}_{t3},\ i = 1, 2$. It follows from above that the two trajectories $\mathbf{y}^{i\alpha_{ti}}$ can be simultaneously generated as the impulse response of the system:

$$\begin{aligned} \zeta_{t+1} &= \mathcal{A}\zeta_t + \mathcal{B}u; \quad u \in R^2 \\ y_t &= \mathcal{C}_t \zeta_t \end{aligned} \tag{15}$$

where

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_L & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_L & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{A}_O & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{A}_L & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{A}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{A}_O \end{bmatrix} \mathcal{B} = \begin{bmatrix} \xi_o^1 & 0 \\ \xi_o^3 & 0 \\ \omega_o & 0 \\ 0 & \xi_o^2 \\ 0 & \xi_o^3 \\ 0 & \omega_o \end{bmatrix},\ \mathcal{C}_L = \begin{bmatrix} \mathbf{I}\, 0 \ldots 0 \end{bmatrix},\ \mathcal{C}_O = \begin{bmatrix} \mathbf{I}\, 0 \ldots 0 \end{bmatrix}$$

$$\mathcal{C}_t = [\alpha_{t1}\mathcal{C}_L\ \alpha_{t3}\mathcal{C}_L\ (\alpha_{t1} - \alpha_{t3})\mathcal{C}_O\ \alpha_{t2}\mathcal{C}_L\ -\alpha_{t3}\mathcal{C}_L\ (\alpha_{t2} - \alpha_{t3})\mathcal{C}_O] \tag{16}$$

Step 2: Recall [25] that for linear time invariant systems, given a triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, with $\mathcal{A} \in R^{n \times n}$, the order of the minimal realization $(\mathcal{A}_m, \mathcal{B}_m, \mathcal{C}_m)$ that has the same input/output response is given by the rank of the product of its controllability and observability matrices, defined as:

$$\mathcal{K}_{ctrb} = \begin{bmatrix} \mathcal{B}\ \mathcal{A}\mathcal{B} \ldots \mathcal{A}^{n-1}\mathcal{B} \end{bmatrix},\ \mathcal{K}_{obs} = \begin{bmatrix} \mathcal{C}^T\ \mathcal{A}^T\mathcal{C}^T \ldots (\mathcal{A}^{n-1})^T\mathcal{C}^T \end{bmatrix} \tag{17}$$

However, this result cannot be directly applied to (15), due to the time–varying scaling factors α_{ti} in $\mathcal{C}_t$. In this case, the order of the minimal realization $(\mathcal{A}_m, \mathcal{B}_m, \mathcal{C}_m)$ that has the same input/output response as the original triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is given by ([25], Chapter 9) $\mathrm{rank}(W_t^c W_t^o)$ where

$$\mathbf{W}_t^o = (\mathcal{K}_{t,o})^T \mathcal{K}_{t,o},\ \mathbf{W}_t^c = (\mathcal{K}_{t,c})^T \mathcal{K}_{t,c},\ \mathcal{K}_{t,o} = \begin{bmatrix} \mathcal{C}_{t-1} \\ \mathcal{C}_{t-2}\mathcal{A} \\ \vdots \\ \mathcal{C}_o\mathcal{A}^{t-1} \end{bmatrix},\ \mathcal{K}_{t,c} = \begin{bmatrix} \mathcal{B}\ \mathcal{A}\mathcal{B} \dots \mathcal{A}^{t-1}\mathcal{B} \end{bmatrix}$$

Note that the pair $(\mathcal{A}, \mathcal{B})$ is time invariant (since no scaling factors are involved). Further, from a PBH argument (see [25], page 366) it can be shown that, if $t \geq n$, then, generically, $\mathrm{rank}(\mathcal{K}_{t,c}) = n$. On the other hand, using the explicit expressions for $\mathcal{A}$ and $\mathcal{C}$ yields, for each block-row of $\mathcal{K}_{t,o}$:

$$(\mathcal{K}_{t,o})_j = \begin{bmatrix} \alpha_{(t-j)1} \left(K_{obs}^L\right)_j & -\alpha_{(t-j)3} \left(K_{obs}^L\right)_j & (\alpha_{(t-j)1} - \alpha_{(t-j)3}) \left(K_{obs}^O\right)_j \\ \alpha_{(t-j)2} \left(K_{obs}^L\right)_j & -\alpha_{(t-j)3} \left(K_{obs}^L\right)_j & (\alpha_{(t-j)2} - \alpha_{(t-j)3}) \left(K_{obs}^O\right)_j \end{bmatrix}$$

where $(\mathbf{M})_j$ denotes the j^{th} block–row of a matrix $\mathbf{M}$, and K_{obs}^L, K_{obs}^O denote the observability matrices of $(\mathcal{C}_L, \mathcal{A}_L)$ and $(\mathcal{C}_O, \mathcal{A}_O)$, respectively. Since by construction both realizations are observable, it follows that, if the motion of O_k has at least one mode not contained in the operator $\mathcal{L}$ (the relative motion of the rigid with respect to O) then the minimum rank of $\mathcal{K}_{t,o}$ over all $\alpha_{ti} > 0$ is achieved by selecting $\alpha_{t1} = \alpha_{t2} = \alpha_{t3} = \alpha_t$, an overall, time varying scaling factor. Further, note that this minimum is achieved by an LTI system if and only if $\alpha_t = \lambda_o \rho^t$ for some $\lambda_o, \rho \neq 0$.

Step 3. Let $\hat{Z}_{ti}$ and $\hat{\mathbf{P}}_{ti}$, denote the actual values of Z_{ti} and the 3D trajectories, respectively. Consider any candidate trajectory $\tilde{Z}_{ti} \doteq \alpha_{ti}\hat{Z}_{ti}$ and denote by $\mathbf{P}_{ti}$, the 3D trajectory reconstructed from the 2D data using $\tilde{Z}_{ti}$. Finally, define the difference vectors:

$$\mathbf{y}_t^i \doteq \mathbf{P}_{ti} - \mathbf{P}_{t3} = \left(\alpha_{ti}\hat{\mathbf{P}}_{ti} - \alpha_{t3}\hat{\mathbf{P}}_{t3}\right) \tag{18}$$

and the associated matrix $\mathbf{H}_y = \left[\mathbf{H}_{\mathbf{y}^1}\ \mathbf{H}_{\mathbf{y}^2}\right]$. Consider any sequence $\tilde{\alpha}_{ti} > 0$ and let $\mathcal{L}(\tilde{\alpha}_{ti})$ denote the associated operator. From step 2 above, it follows that

$$min_{\alpha_{ti}} \mathrm{rank}\{\mathcal{L}(\alpha_{ti})\} \leq \mathrm{rank}\{\mathcal{L}(\tilde{\alpha}_{ti})\} \leq \mathrm{rank}\{\mathbf{H}(\tilde{\alpha}_{ti})\}$$

with the equalities holding only in the case where $\mathcal{L}$ is an LTI operator, e.g. $\tilde{\alpha}_{ti} = \lambda_o \rho^t,\ i = 1, 2, 3$. Hence, the depths Z_{ti} obtained by minimizing the rank of $\mathbf{H}_y$ satisfy $Z_{ti} = \lambda_o \rho^t \hat{Z}_{ti}$ for some $\lambda_o, \rho \neq 0$.