Practical Projective Structure from Motion (P^{2}SfM)

Ludovic Magerand, Alessio Del Bue
Visual Geometry and Modelling (VGM) Lab, Istituto Italiano di Tecnologia (IIT)
Via Morego 30, 16163 Genova, Italy
ludovic@magerand.fr, Alessio.DelBue@iit.it

Abstract

This paper presents a solution to the Projective Structure from Motion (PSfM) problem able to deal efficiently with missing data, outliers and, for the first time, large scale 3D reconstruction scenarios. By embedding the projective depths into the projective parameters of the points and views, we decrease the number of unknowns to estimate and improve computational speed by optimizing standard linear Least Squares systems instead of homogeneous ones. In order to do so, we show that an extension of the linear constraints from the Generalized Projective Reconstruction Theorem can be transferred to the projective parameters, ensuring also a valid projective reconstruction in the process. We use an incremental approach that, starting from a solvable sub-problem, incrementally adds views and points until completion with a robust, outliers free, procedure. Experiments with simulated data shows that our approach is performing well, both in term of the quality of the reconstruction and the capacity to handle missing data and outliers with a reduced computational time. Finally, results on real datasets shows the ability of the method to be used in medium and large scale 3D reconstruction scenarios with high ratios of missing data (up to 98%*).*

Notation. Homogeneous coordinates of a vector $\mathbf{v}$ are written as $\tilde{\mathbf{v}} = [\mathbf{v}^\top 1]^\top$ and $\mathtt{I}_{m\times n}$ is the $m \times n$ identity matrix. A $m \times n$ matrix of 0 (or 1) is denoted $0_{m\times n}$ (or $1_{m\times n}$) and $\mathbf{0}_n$ (or $\mathbf{1}_n$) is a n-vector of 0 (or 1). Symbols $\odot$ and $\otimes$ are used for the element-wise and tensor product respectively. The Moore-Penrose pseudo-inverse of a real vector $\mathbf{v}$ is written $\mathbf{v}^+ = \mathbf{v}^\top/\|\mathbf{v}\|^2$ and its associated skew symmetric matrix is noted $[\mathbf{v}]_\times$.

1. Introduction

Robust factorization methods have been highly successful in delivering a solution to affine Structure from Motion (SfM) even in the presence of large amounts of missing data and outliers [17, 24]. However, the Projective Structure from Motion (PSfM) [26] problem still entails difficulties and despite considerable efforts, there are clear limitations in current approaches [22, 6, 31, 13, 5, 18, 19, 15, 8, 9, 14]. These problems span from the non-linearity given by the perspective camera model to the relevant presence of missing data, noise and outliers in the measurement matrix containing the 2D observations. These nuisances have restricted the applicability of PSfM to relatively small 3D reconstruction scenarios with few points and small percentages of missing data. Differently, this paper shows how PSfM can be solved for challenging real datasets by lessening the non-linearities of previous approaches.

In detail, given f images of a scene and correspondences between a set of n image points in multiple-views, SfM estimates the 3D position of each point and the camera poses. The simplest instance of SfM adopts affine cameras for 3D projection that leads to a bilinear model in the form of: $\mathsf{M} = \mathsf{PS}$. The measurement matrix M (of size $3f \times n$) contains the homogeneous image projections $\tilde{\mathbf{m}}_{i,j}$ while P (of size $3f \times 4$) represents the vertical concatenations of the 3×4 camera matrices P_i and S (of size $4 \times n$) is the horizontal concatenation of the homogeneous 3D points $\tilde{\mathbf{s}}_j$. As M is resulting from a product of fixed size matrices, a rank-4 constraint exists and it has been used in [28] to factorize such matrix into (P, S) up to an ambiguity using standard computational tools (*e.g.* Singular Value Decomposition – SVD). This factorization approach to the SfM problem has been successfully applied to obtain a global solution, meaning that all the data is used at once, and usually providing closed-form solution without the need of an initialisation.

However the affine model restricts applicability to specific scenarios while current challenges in computer vision go towards reconstructing large scenes where the assumptions of affine cameras are no longer satisfied. Upgrading the camera model to perspective leads image projections that also depend on the 3D points depths with respect to the camera, resulting in a different problem defined as:

$$\mathsf{M} \odot (\mathsf{D} \otimes \mathbf{1}_3) = \mathsf{P}\,\mathsf{S}, \tag{1}$$

where D is a $f \times n$ matrix containing the projective depths for all projections.

Moreover, when dealing with images having wide baselines, it is rather common to have 3D points occluded either by the scene itself or because being out of the camera field. As a consequence, the matrix M is often incomplete with some of its entries missing. Completing these entries leads to an NP-hard problem [21, 10] that can be defined as:

$$(\mathsf{Z} \otimes \mathbf{1}_3) \odot \mathsf{M} \odot (\mathsf{D} \otimes \mathbf{1}_3) = (\mathsf{Z} \otimes \mathbf{1}_3) \odot (\mathsf{P}\,\mathsf{S}), \qquad (2)$$

where the $f \times n$ binary matrix Z indicates the known entries.

These missing correspondences can also result from failures in matching the image projections of the 3D points. Mismatches or extremely noisy correspondences can also be present and are usually referred to as outliers. Once detected, they can be removed by nullifying the corresponding entries in Z. The presence of the projective depths D, missing data and outliers are all aspects that have to be dealt with in order to provide a successful method for PSfM.

1.1. Related Work

Tomasi and Kanade [28] proposed the first factorization based approach to SfM using orthographic cameras without missing data. A first estimate of the low-rank bilinear components was obtained through SVD and afterwards a metric correction based on constraints raising from the orthographic camera model was used to recover the 3D structure solely from image trajectories. Considering multi-view geometry relations, Sturm and Triggs [26, 30] proposed the first extension to perspective cameras by finding a projective depths matrix D which allows the SVD to factorize $\mathsf{M} \odot (\mathsf{D} \otimes \mathbf{1}_3)$ as a product of two rank 4 matrices. To compute D, pairwise fundamental matrix estimations were linked together, which can result into accumulation of errors. Moreover, [22] showed that this method can sometime converge to useless results.

There have been several attempts to improve Sturm and Triggs solution [22, 6, 31, 13, 5, 18, 19, 15, 8, 9] providing, in most of the cases, iterative methods given the non-linearity of the problem. Instead of using pairwise relations to compute D, such local iterative approaches usually start initializing $\mathsf{D} = 1_{f\times n}$ and then adjusting D using the rank constraint while optimizing the reprojection error. These approaches differ mainly by the constraints used to prevent the convergence to trivial and ill-conditioned solutions, except for [8] which proposes a SDP formulation based on a trace norm minimization making it suitable for global optimization. Only few of the mentioned methods [19, 15, 8, 9] try to tackle projective reconstruction with missing data. Recently, Hong *et al*. [14] presented a projective bundle adjustment method based on a Variable Projection approach from an arbitrary initialization. Convergence to trivial solutions is prevented by a penalty term discouraging update along the column space of the initial P.

Given previous attempts to solve the problem, it was becoming clearer that more attention needed to be posed on the constraints over D. Using multi-view geometry considerations, Nasihatkon *et al*. [20] rightly pointed out in the Generalized Projective Reconstruction Theorem (GPRT) that only specific configurations of the projective depths matrix D can provide a solution leading to a correct 3D reconstruction. Except for [8], the previous methods mentioned above do not comply fully with this theorem.

Online or incremental methods for matrix factorization are a preferable choice when the data matrix is of considerable size as shown by [17, 16] in the case of affine SfM, especially to handle outliers [24]. However, such a solution, computationally viable and reconstruction friendly, is still not available for PSfM in the literature and in the next section we will show our contributions to this end. Our approach is related to [24] limited to Henneberg constructive extensions and adapted to the perspective case.

1.2. Proposed Approach and Contributions

First, we make explicit that (P, S) already contain the projective depths information thus being useless to re-estimate such parameters as done in most previous approaches. This results in a more **compact parametrisation** of the PSfM problem which is still bilinear. Moreover, we show that a generalisation of the step-like mask constraint on the projective depths of [20] can be linearly transferred to the projective estimation of (P, S). This leads to **efficient optimization** based on alternating simple standard linear Least Squares minimizations.

Then, similarly to the affine case [24], our method adopts an incremental procedure to solve the PSfM problem. This strategy is key to success in the presence of outliers and high ratio of missing data since it allows to select parts which are solvable through a robust, RANSAC-based, fitting procedure to **remove outliers** which are then treated as missing data. In this regard, we demonstrate, for the first time, that PSfM can deal with large scale scenarios typical of the most advanced bundle adjustment based pipelines [25].

2. Compact Factorization Based Formulation

We now present in this section a formulation of the PSfM problem where the projective depths are eliminated. This leads to the core linear Least Squares systems that are the building blocks for our incremental efficient and robust pipeline to solve the PSfM problem.

2.1. Projective Parameters Fundamental Relations

Let X and Q be the respective estimation of P and S up to a 4×4 invertible projective ambiguity Y meaning $\mathsf{X} = \mathsf{P}\mathsf{Y}$ and $\mathsf{Q} = \mathsf{Y}^{-1}\mathsf{S}$. An estimation of the projective parameters of a point $\mathbf{s}_j$ (or a camera P_i) is then the 4-vector $\mathbf{q}_j$

corresponding to the column j of Q (or the 3×4 matrix X_i corresponding to rows $3i-2$ to $3i$ of X). The fundamental relation between the projective parameters X_i and $\mathbf{q}_j$, the projective depth $d_{i,j}$ of point j in view i and the 2D projection $\mathbf{m}_{i,j}$ is given by

$$d_{i,j}\tilde{\mathbf{m}}_{i,j} = \mathsf{X}_i \mathbf{q}_j. \tag{3}$$

Having an estimation of the projective parameters X_i and $\mathbf{q}_j$, it results that the projective depth can be estimated as

$$d_{i,j} = \tilde{\mathbf{m}}_{i,j}^{+} \mathsf{X}_i \mathbf{q}_j. \tag{4}$$

Eliminating the projective depths $d_{i,j}$ from Eq. (3) can be done as in the DLT method [12] using the cross product resulting in

$$\mathsf{E}\left[\tilde{\mathbf{m}}_{i,j}\right]_{\times} \mathsf{X}_i \mathbf{q}_j = \mathbf{0}, \tag{5}$$

where E is a 2×3 matrix containing the two first rows of the identity and is used to remove the linear dependency between the third line and the first two.

Note that DLT leads to minimizing the algebraic error and, following [12], an appropriate normalization of the data is necessary and introduced in Sec. 3. Another elimination method can be obtained using Eq. (4) to substitute the projective depths in Eq. (3). While this provides a MLE similar to the reprojection error, it seems less accurate experimentally[1].

Assuming we know the projective parameters of v views where the image projections of the point j are visible, the corresponding projective parameters $\mathbf{q}_j$ must satisfy

$$\begin{bmatrix} \mathsf{E}\left[\tilde{\mathbf{m}}_{1,j}\right]_{\times} \mathsf{X}_1 \\ \vdots \\ \mathsf{E}\left[\tilde{\mathbf{m}}_{v,j}\right]_{\times} \mathsf{X}_v \end{bmatrix} \mathbf{q}_j = \mathbf{0}_{2v}. \tag{6}$$

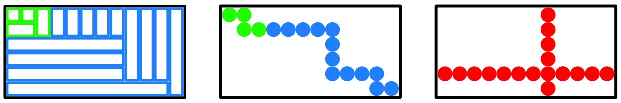
If the view i contains the image projections of p points for which estimations of their projective parameters are available, then its projective parameters X_i vectorized row by row as $\mathbf{x}_i$ are such that

$$\begin{bmatrix} \mathsf{E}\left[\tilde{\mathbf{m}}_{i,1}\right]_{\times} \mathsf{G}_1 \\ \vdots \\ \mathsf{E}\left[\tilde{\mathbf{m}}_{i,p}\right]_{\times} \mathsf{G}_p \end{bmatrix} \mathbf{x}_i = \mathbf{0}_{2p}\,, \quad \mathsf{G}_j^{\top} = \begin{bmatrix} \mathbf{q}_j & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{q}_j & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{q}_j \end{bmatrix}. \tag{7}$$

These two systems are homogeneous and linear in either $\mathbf{q}_j$ or $\mathbf{x}_i$. They can be written generically as

$$\mathsf{A}\mathbf{y} = \mathbf{0}, \tag{8}$$

where A is of size $2v \times 4$ (point case) or $2p \times 12$ (view case).



(a) Tiles, each one has norm or sum of some elements fixed.

(b) Step-like mask, the dots are the fixed entries.

(c) Cross-shaped matrix, the dots are the only non-null entries.

Figure 1. The black rectangular boxes represent the matrix D containing the projective depths for 6 cameras (rows) and 12 points (columns). Dots represent single entries while small boxes are tiles that contain possibly more than one entry. (b) and (a) show examples of valid constraints. (c) is an invalid configuration of D.

2.2. Projective Parameters Constraints

In this section, we propose a new set of linear constraints on the projective parameters which satisfy the conditions to be reconstruction friendly with respect to the GPRT [20]. This theorem states that D must be diagonally equivalent to the true depth matrix and satisfy the following conditions: no null column or row and not cross-shaped, meaning a null matrix except for a cross as in Fig. 1(c).

Using the same tiling as in Fig. 1(a), for each tile we constrain the projective parameters of the corresponding point or view to be estimated such that

$$\underbrace{\left(\frac{1}{k_v}\sum_{i=1}^{k_v} \tilde{\mathbf{m}}_{i,j}^{+} \mathsf{X}_i\right)}_{=\mathbf{c}^{\top}} \mathbf{q}_j = 1 \quad \text{or} \quad \underbrace{\left(\frac{1}{k_p}\sum_{j=1}^{k_p} \tilde{\mathbf{m}}_{i,j}^{+} \mathsf{G}_j\right)}_{=\mathbf{c}^{\top}} \mathbf{x}_i = 1. \tag{9}$$

Note that the measurements must be available for all the projection considered into this sum. However we do not have to necessarily consider all the visible projections[1].

These constraints can then be used to substitute one of the projective parameters in Eq. (6) or Eq. (7), resulting in a standard linear system which is faster to solve. Doing the substitution to remove y_1, the first entry of $\mathbf{y}$ in Eq. (8), we have to split $\mathbf{A}$, $\mathbf{y}$ and $\mathbf{c}$ as

$$\mathsf{A} = \begin{bmatrix} \mathbf{a} & \mathsf{A}' \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \mathbf{z} \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \mathbf{c}' \end{bmatrix}, \tag{10}$$

where $\mathbf{a}$ and A' are the first and remaining columns of A. Then after the substitution, we need to minimize

$$\mathsf{B}\mathbf{z} = \mathbf{b} \qquad \text{with} \quad \begin{cases} \mathsf{B} = \mathsf{A}' - \mathbf{a}\mathbf{c}'^{\top}/c_1 \\ \mathbf{b} = -\mathbf{a}/c_1 \end{cases}, \tag{11}$$

for the unknown vector $\mathbf{z}$ which is then a minimal parametrization of the projective parameters that contains only three degrees of freedom for a point and eleven for a view. The resulting minimal or overdetermined linear system can be solved or minimized efficiently in the Least Squares sense after which we can retrieve y_1 as

$$y_1 = \frac{1}{c_1}\left(1 - \mathbf{c}'^{\top}\mathbf{z}\right). \tag{12}$$

[1] Check the supplementary material for more details.

From Eq. (4), our constraints can be transferred to the projective depths. When the sum contains only the last element of each tile, they are actually equivalent to the step-like constraints presented in [20] and illustrated in Fig. 1(b), which corresponds to the tiling of Fig. 1(a). This generalization was required as the projection of the last element of each tile is not always visible and it has the advantage of using all the data to build the constraints. To prevent cross-shaped degeneracies, we impose in Sec. 3.2 the first tiles to contain fixed entries forming a 2×3 tetris step-like block coloured green in Fig. 1. As this sub-block cannot be cross-shaped, the final reconstruction cannot be either.

3. Practical Projective SfM (P^{2}SfM)

We describe here our approach to estimate the projective factors X and Q minimizing $\sum_{i,j\in\mathcal{Z}} \|\mathsf{E}\left[\tilde{\mathbf{m}}_{i,j}\right]_\times \mathsf{X}_i \mathbf{q}_j\|_2^2$ subject to the constraints of Eq. (9). A graphical illustration is provided in Fig. 2 and important details on each step are given from Sec. 3.2 to 3.5.

3.1. Overview of the Proposed Method

Before starting, the image projections are normalized to improve the conditioning of the linear Least Squares systems to be solved[2]. It is more computationally efficient to compute all $\tilde{\mathbf{m}}_{i,j}^+$ and $\mathsf{E}\left[\tilde{\mathbf{m}}_{i,j}\right]_\times$ only once and store them into sparse data matrices. The method then starts with an initial sub-problem (Sec. 3.2) and iterates by robustly adding missing tiles (Sec. 3.4) where each tile corresponds to either a view (3-rows) or a point (a column). Multiple views or points can be added at the same time and the procedure continues until no further tile can be added. Searching for tiles to be added depends on the number of visible projections and eligibility thresholds which are dynamically adjusted (Sec. 3.3). After each inclusion, the reconstruction is refined by re-estimating all the points and views already added (Sec. 3.5). The complete method is then given in Alg. 1. The result is a normalized projective reconstruction satisfying the GPRT [20] and the set of inlier projections.

3.2. Initial Sub-Problem Selection and Estimation

The initial sub-problem can be of arbitrary size but in general it is preferable to start from minimal configurations. In such case, we need to find a set of frames and points, *i.e.* a matrix sub-block as in Fig. 2(a) that can be robustly solved to get a valid initial projective reconstruction. This is done in the standard way with robust fundamental matrix estimation [12] after selecting two views using the pyramidal score from [25] in a way similar to the affinity score of [27]. If by chance the robust estimation of the fundamental matrix fails, we move to the next higher score until a solvable sub-matrix is found.

Algorithm 1: Practical Projective SfM (P^{2}SfM).

```
Normalize projections and compute data matrices ;
Find an initial sub-problem and robustly solve it, see sec. 3.2 ;
while reconstruction is not complete and (reconstruction was extended or an eligibility threshold can be decreased) do
    Find currently eligibles views, see sec. 3.3 ;
    Try to add eligibles views robustly, see sec. 3.4 ;
    if at least one eligible view has been added then
        Increase the eligibility threshold for points ;
        Refine locally the reconstruction, see sec. 3.5 ;
    else if no view was eligible then
        Decrease the eligibility thresholds for views if not minimum ;
    Find currently eligibles points, see sec. 3.3 ;
    Try to add eligibles points robustly, see sec. 3.4 ;
    if at least one eligible point has been added then
        Increase the eligibilty thresholds for views ;
        Refine locally the reconstruction, see sec. 3.5 ;
    else if no point was eligible then
        Decrease the eligibility threshold for points if not minimum ;
Refine globally the reconstruction, see sec. 3.5 ;
```

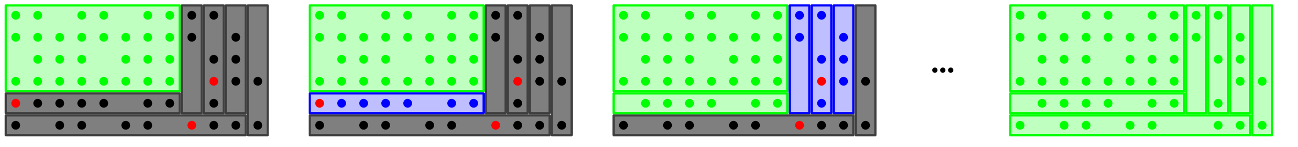
After extracting the epipolar geometry from the estimated fundamental matrix as in [26, 30], an SVD of the sub-matrix can be used to compute the projective parameters of the initial two views and the inlier points. The resulting projective parameters are then balanced to match the constraints as defined in Sec. 2.2.

3.3. Finding Eligibles Views and Points

Finding the views or points to add next is a critical issue. In order to do so, we first define a point or view as *known* if an estimation of its projective parameters is available. Initially known points and views are therefore given by the solution of the initial sub-problem. We then call a point (or view) *eligible* if there are more visible projections in known views (or points) than a given eligibility threshold, which is different for points and views. For views, we also compute the pyramidal score [25] of the visible known points and reject them if below a threshold.

If the eligibility thresholds are set too high, which is desirable as it usually gives better estimations, it might happen that no point or camera is eligible. In order to limit premature interruptions of the algorithm, the eligibility thresholds are dynamically adjusted in Alg. 1 between an initial high value and a minimum value both provided by the user. We also included a rejection mechanism for points or views that

[2] Details on this step are given in the supplementary materials.



(a) Initial sub-problem (green).
(b) Adding one view (blue).
(c) Adding three points (blue).
(d) Final reconstruction and inliers.

Figure 2. Example of the incremental procedure to reconstruct a scene with 6 views and 12 points. The dots indicates visible projections, red ones are outliers. We start in (a) with a previously solved sub-problem of 4 views and 8 points in green, grey tiles indicates data not yet considered. Then at each step, green tiles are the current reconstruction and tiles currently added to expand it are in blue. For instance, in (b) we robustly add a view, automatically removing an outlier projection. In (c) we then robustly add three points. This is repeated until we reach a final outliers free reconstruction in (d).

previously failed the robust estimation (see next section). As a consequence, another condition to be eligible is that the number of visible projections is greater than when the last failure happened.

3.4. Robustly Adding a Point or View

Our method is based on minimizing the linear Least Squares systems of Eq. (11) to estimate the projective parameters, which are known to be sensible to outliers due to mismatches or strong noise. To deal with this, the estimation is done robustly using a Locally Optimized RANSAC [7] with the MSAC score [29] and an adaptive stopping criterion given a minimum confidence of finding the optimal inlier set. Projections detected as outliers are then removed from the measurements matrix and treated as missing data.

During this procedure, we reject any random subset leading to a rank deficient B, a bad condition number of B or an excessive error in Eq. (11). The two first cases can happen with degenerate configurations of points or views but more frequently when estimating a view [11]. In order to get better estimation from the random subset, we also increased its size. After selecting the inlier set of the visible projections by using a threshold on the reprojection errors, we prune projections for which the projective depth is negative or null. Finally we also reject the estimation if the resulting inlier set is smaller than the random subset.

If no correct estimation can be found before a given maximum number of iterations, we temporary reject the view or point. When new projections will be available for this view or point, we try to add it again, ensuring the random subsets contain at least one of the new projections. This is necessary as a complete random subset would most likely contains only previously rejected projections if there are just a few new projections. The estimation would then fail as they have already been through this procedure once.

3.5. Reconstruction Refinement

Because an incremental procedure does not consider all the information at once, it can be affected by errors accumulation while iterating. To prevent this, we refine the overall reconstruction after trying to add eligible points (or views) if any addition was successful. The refinement is done by alternating new estimations of all the projective parameters, starting from views (or points), and continue until the overall change in the projective parameters is small enough. This is done without the robust procedure but using only projections previously accepted as inliers. While re-estimating, we use the same visible projections to build the constraints as when the points or views were first added. A similar method was proposed in [5] but without any constraints on the projective depths. To speed up the process while doing the completion, the refinement is done only over the views and points added in the two last iterations of the main loop. The last refinement in line 18 of Alg. 1 provides the final reconstruction and it is done over all the estimated points and views.

4. Experimental Results

We validated the practicability of our approach with both synthetic and real experiments evaluating performance in realistic cases with high percentages of missing data and outliers. We compared our method (P^{2}SfM) with [9] (YDHL) and [14] (VarPro) that consistently outperform previous works thus making adequate the comparison with these methods only.

4.1. Synthetic Dataset Results

To evaluate the proposed approach, 100 simulated sequences were generated with a missing data pattern that models points falling out from the cameras field of view as it is advisable to avoid randomly removed matches [2]. For each sequence, the 3D shape was obtained by randomly generating 200 points inside a cube of unit dimension. A set of 15 cameras was simulated from random intrinsic and extrinsic parameters inside realistic ranges. Cameras were placed randomly in a 1.25 units cube, looking at a random position inside a 0.8 unit cube. Focal lengths are drawn from $[1500; 30500]$ pixels and sensor widths range from 800 to 6800 pixels with a 1.33 or 1.5 aspect ratio. We ensured that each point was seen at least in four views and each view contained at least 18 points projections.

To achieve exactly the tested ratios (from 50% to 75%), we removed very few random entries when necessary. Finally, noise was simulated with a centred Gaussian on each visible image projection. For evaluating results, the 3D error on one sequence is calculated as $\left\|\mathsf{S}-\mathsf{S}^{GT}\right\|_F / \left\|\mathsf{S}\right\|_F$ after registering the estimated 3D points with Procrustes analysis. The 2D error is computed as the root mean square (rms) of all the reprojection errors. All errors are then averaged over all the sequences of the dataset.

4.1.1 Robustness to Outliers

For this experiment, we generated up to eight outliers by replacing randomly some projections with random coordinates inside the corresponding views. While all methods have a very small reprojection error without outliers, even one is enough to decrease drastically the performance of previous works as it can be observed in Fig. 3(a). This impacts also the 3D points reconstruction error which grow quickly for them in Fig. 3(b). Differently, our approach shows strong resilience to increasing number of outliers.

Note that for a given sequence, previous works return a result for all points and views or for none of them while our method always gives a reconstruction where some points or views might be unestimated due to the rejection mechanism of Sec. 3.4. In Fig. 3(c), the entire synthetic dataset is considered and the percentage of unestimated points corresponds to the number of failed sequences for YDHL and VarPro and the cumulative unestimated points for our method. We see that VarPro fails less often than YDHL and is not afflicted much by a few outliers. P^{2}SfM is unaffected at all by outliers, the small percentage of unestimated points is constant and induced by noise.

4.1.2 Running Time Comparisons

Running times were obtained on a laptop having an intel core i7-6700HQ processor and 16GB memory. No outliers were added and the missing data ratio was kept to 60%. Fig. 4(a) shows all algorithms performance on small scale datasets of growing size. For each dataset size, ten sequences were run and the average time is given. Both VarPro and P^{2}SfM are way faster than YDHL, clearly demonstrating that including the projective depths as parameters of the problem is computationally expensive.

When dealing with medium scale sequences, Fig. 4(b)(c) show the behaviour when increasing the number of points and views respectively. For each size, the running time is averaged over five sequences. The online procedure and LLS minimization are keys to reduce computational costs compared to VarPro. Due to high memory usage, YDHL could not be run on the three biggest sequences and the three smallest gave no result after 4 hours of computation.

Notice that all methods have been implemented in MATLAB[3] with no parallelization involved except for subroutines natively supporting it. Using another language and the shared memory paradigm, P^{2}SfM can be massively accelerated by estimating points (or views) in parallel.

4.1.3 Missing Data and Noise Effect

Fig. 5(a) shows the evolution of the 2D reprojection error when increasing the noise level from a 0 to 2 pixel standard deviation at two ratios of missing data (55% and 70%). Both VarPro and P^{2}SfM outperform YDHL even in the noise-free case where they achieve an almost perfect reconstruction. They have similar behaviour for noise growing up to 1 pixel, and then the robust estimation of P^{2}SfM starts filtering projections with high noise resulting in a decreased error.

The behaviour of the 3D structure error is displayed on Fig. 5(b) when the missing data ratio grows from 50% to 75% in presence of noise ($\sigma = 0.5$ or 1.5 pixels). With a low noise, VarPro and P^{2}SfM have a similar evolution with low errors. Due to the limited size of the sequences, when the noise is higher and projections are filtered by the robust estimation, few data remain available to the LLS estimation in P^{2}SfM and results in an higher error. In both cases, YDHL achieves the lowest accuracy.

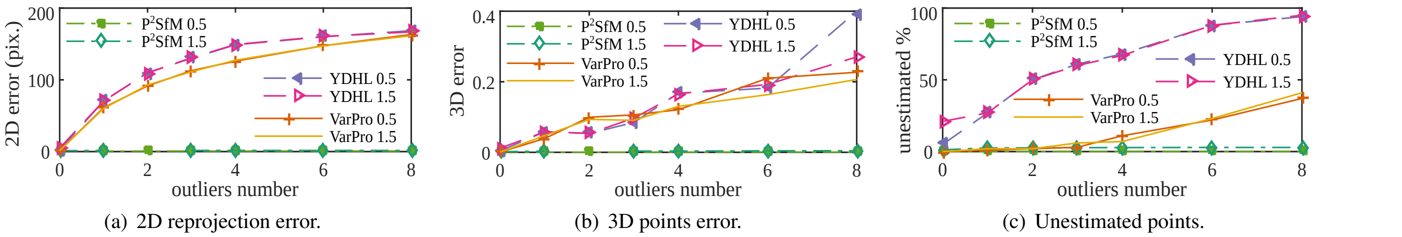
4.2. Real Data

In Tab. 1, we also evaluated P^{2}SfM on various real datasets of different size available in the literature. When necessary, feature extraction and matching have been done off-line once for all methods prior to reconstructions using the first stage of COLMAP [25] and we built the measurement matrix from the output. To obtain an Euclidean reconstruction from the projective one, we used the metric upgrade method of [4]. Given timings do not include the time required for all these steps. Further results are provided in the supplementary materials.

4.2.1 Small Scale Sequences

Six small scale sequences containing less than a million entries in the measurements matrix M were evaluated and results are given in Tab. 1. P^{2}SfM outperforms VarPro and YDHL on both the 2D rms reprojection error and the running time for all sequences. An example of the 3D reconstruction obtained is given on Fig. 6(b) for the famous Dino sequence. It shows a dinosaur toy being rotated in front of a camera, which results in elliptical trajectories for the completed measurements as seen on Fig. 6(a). We used the 4983 points experiment since the smaller Dino sequence is mostly suited for affine structure from motion approaches [3] and it has a low missing data ratio.

[3] Our implementation is freely available online at https://bitbucket.org/lmagerand/ppsfm.

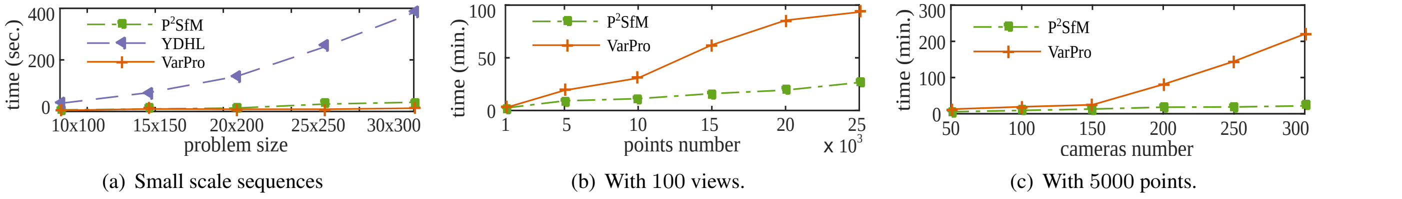


(a) 2D reprojection error.

(b) 3D points error.

(c) Unestimated points.

Figure 3. Behavior with outliers for the reprojection error (a), the 3D structure error (b) and the number of unestimated points (c) at two different standard deviations of the noise ($\sigma = 0.5$ and $\sigma = 1.5$ pixels) and 60% of missing data. YDHL and VarPro are quickly and strongly afflicted by outliers while P^{2}SfM is almost unaffected thanks to the RANSAC based estimation.

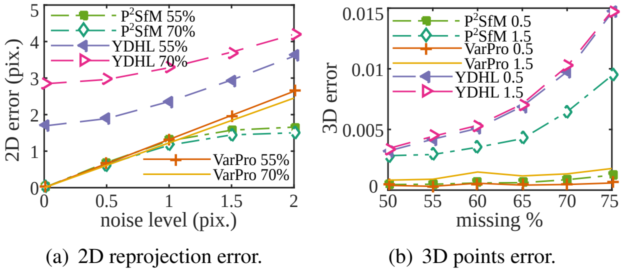


(a) Small scale sequences

(b) With 100 views.

(c) With 5000 points.

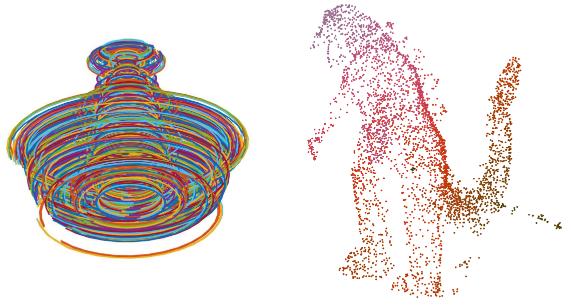
Figure 4. Running times of P^{2}SfM compared to YDHL and VarPro on small scale sequences (a). On the larger scale experiments, timings are reported with increasing number of points (b) or views (c) for P^{2}SfM and VarPro. If on small scale sequences VarPro is faster (a), P^{2}SfM has a clear advantage on larger scale sequences (b)(c). YDHL is the slowest and cannot even handle medium scale sequences.



(a) 2D reprojection error.

(b) 3D points error.

Figure 5. Noise level effect on the 2D reprojection error (a) and behaviour of the error on 3D structure (b) with increasing missing data ratio. Both VarPro and P^{2}SfM outperform YDHL.



(a) Completed measurements.

(b) 3D reconstruction.

Figure 6. The dinosaur sequence. (a) shows the 2D image trajectories after completion with a random colour for each one, making evident the rotational motion of the dino. (b) presents the 3D reconstruction after metric upgrade where the colours gradient corresponds to the depth along the reconstruction principal axis.

4.2.2 Medium and Large Scale Sequences

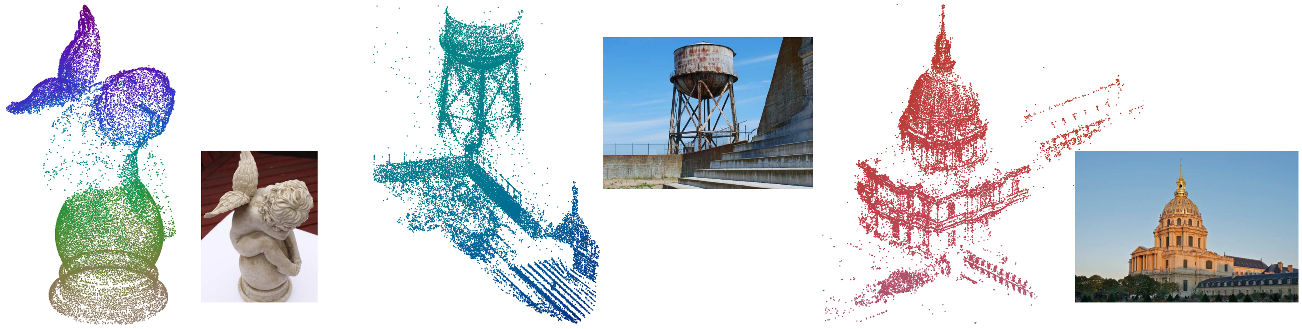
As seen in Tab. 1 (the last four rows), existing PSfM approaches are unable to reconstruct any of the medium or large scale sequences evaluated which contain millions of entries in the measurements matrix M. VarPro could not complete them before exhausting available memory or reaching a twelve hours time limit. YDHL is already having troubles processing some of the small scale sequences and was not evaluated here. Differently, our method successfully delivers correct reconstructions, making it the first PSfM method able to deal with such datasets. We compared our results to COLMAP [25], a standard bundle adjustment based method implemented in C++ using highly optimized libraries and a camera model with radial distortion.

The first sequence consists of high resolution images of a cherubim statue [1]. The scene displayed in the second and third one are parts of Alcatraz, showing a corner of the courtyard with its water tower and the west side of the main building [23]. The feature points for these three sequences have been extracted and matched using the first stage of COLMAP. The last sequence is presented in [23] and it is taken around the Dome des Invalides in Paris.

A view of the corresponding 3D reconstructions are given in Fig. 7 and Fig. 8 (the 3D reconstructed models are available in the supplementary materials). While COLMAP achieves a lower reprojection error, we need about half the time to process the two largest sequences. Given the size of the Alcatraz West Side sequence (about 120 millions entries in M), to the best of our knowledge this is the largest successful test for a PSfM method.

Sequence			P^{2}SfM		VarPro [14]		YDHL [9] or COLMAP [25]		
Name	Size	Missing	2D error	Time (sec.)	2D error	Time (sec.)	2D error	Time (sec.)	
House (VGG)	10 × 672	57.7%	0.4268	2.12	0.6246	68.9	0.6639	1054	YDHL
Corridor (VGG)	11 × 737	50.2%	0.3626	4.11	0.3853	331	0.4329	978	
Dinosaur 319	36 × 319	76.9%	0.4652	4.12	1.5761	90.8	3.3543	609	
Dinosaur 4983	36 × 4983	90.8%	0.3477	27	1.6492	1428	Time Limit (4H)		
Wilshire (Ponce)	190 × 411	60.7%	0.5011	52	0.5995	3623	0.6688	3851	
Blue Teddy Bear (Ponce)	196 × 827	80.7%	0.6067	385	1.4169	13341	Time Limit (4H)		
Cherubim [1]	65 × 72785	93.3%	0.9136	234	Time Limit (12H)		0.4827	182	COLMAP
Alcatraz Courtyard [23]	173 × 59488	94.5%	0.7233	579	Out of Memory (8GB)		0.4226	1696	
Alcatraz West Side [23]	435 × 138811	98.4%	0.9124	2807	Out of Memory (8GB)		0.5728	5141	
Dome des Invalides [23]	85 × 94939	91.9%	0.3812	451	Out of Memory (8GB)		Not Available (no images)		

Table 1. Real sequences results. P^{2}SfM, VarPro and YDHL were evaluated on six small scale sequences. The results confirm what is observed on the synthetic dataset, VarPro and P^{2}SfM outperform YDHL. On medium scale, P^{2}SfM was evaluated on four sequences against VarPro and COLMAP. VarPro could not provide a result for these sequences while COLMAP is usually slower than P^{2}SfM.



(a) Cherubim. (b) Alcatraz Courtyard. (c) Dome des Invalides.

Figure 7. Reconstructions obtained with P^{2}SfM for the medium scale sequences. All P^{2}SfM reconstructions are convincing with respect to the scene observed. The colours gradient corresponds to the depth along the reconstruction principal axis.

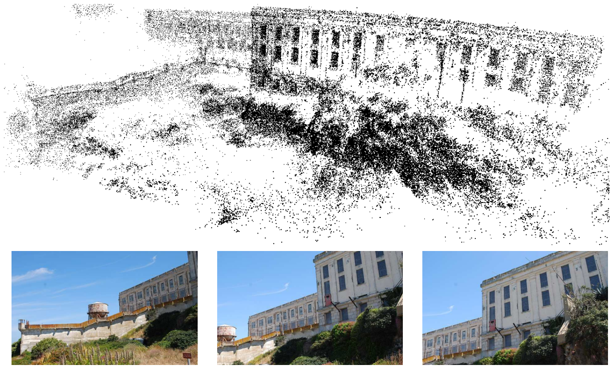


Figure 8. The Alcatraz West Side sequence reconstructed using P^{2}SfM and three sample images over the 435. This is a reasonable reconstruction of 138811 points obtained in only 47 minutes.

4.3. Implementation Details

In order to obtain best results, we recommend setting the minimum eligibility thresholds to at least 18 for views and 4 for points whenever possible, and advise to not set them below 12 and 3 respectively. We used initials values of 48 for views and 12 for points. The parameters for the robust estimation were: an outlier threshold of 4 pixels, a maximum number of iterations fixed at 5000, and a confidence of 99.99% of having found the optimum set on early exit. During the factorization, the refinement is halted when the change in the projective parameters is less than 10^{-7} or 50 iterations have been done. The final refinement is made with a threshold of 10^{-8} over the parameters change or a maximum of 150 iterations.

5. Conclusion

This paper presented P^{2}SfM, an efficient method to solve the PSfM problem in the case of strong ratios of missing data and relevant outliers corrupting the measurements. Constraints has been included to comply with the GPRT, ensuring a correct projective reconstruction. The method was tested against challenging real scenarios with up to 98% missing data ratio and it has shown comparable or better performance with respect to previous PSfM approaches, making it a practical PSfM method. Future work will be dedicated in adapting the method to hierarchical approaches such as [27, 24]. To improve error containment, a global non linear refinement will be integrated. Detecting and merging similar points tracks as COLMAP could also increase the quality of the reconstruction. Finally, to further improves efficiency, a parallelized implementation is also considered.

References

[1] 3Dflow SRL. 3DF Zephyr reconstruction showcase. http://www.3dflow.net/, 2016. 7, 8

[2] S. Bhojanapalli and P. Jain. Universal matrix completion. In *The 31st International Conference on Machine Learning (ICML 2014)*, 2014. 5

[3] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 2, pages 316–322, 2005. 6

[4] M. Chandraker, S. Agarwal, F. Kahl, D. Kriegman, and D. Nister. Autocalibration via rank-constrained estimation of the absolute quadric. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, Minneapolis, 2007. 6

[5] Q. Chen and G. Medioni. Efficient iterative solution to m-view projective reconstruction problem. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 2, 1999. 1, 2, 5

[6] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(11):1098–1104, 1996. 1, 2

[7] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *DAGM-Symposium*, pages 236–243, 2003. 5

[8] Y. Dai, H. Li, and M. He. Element-wise factorization for n-view projective reconstruction. In *European Conference on Computer Vision (ECCV)*, volume 6314 of *Lecture Notes in Computer Science*, pages 396–409, 2010. 1, 2

[9] Y. Dai, H. Li, and M. He. Projective multiview structure and motion from element-wise factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2238–2251, Sept 2013. 1, 2, 5, 8

[10] N. Gillis and F. Glineur. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011. 2

[11] R. Hartley and F. Kahl. Critical configurations for projective reconstruction from multiple views. *International Journal of Computer Vision*, 71(1):5–47, 2007. 5

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2003. 3, 4

[13] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13):981–991, November 1999. 1, 2

[14] J.-H. Hong, C. Zach, A. Fitzgibbon, and R. Cipolla. Projective bundle adjustment from arbitrary initialization using the variable projection method. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 5, 8

[15] H. Jia and A. M. Martinez. Low-rank matrix fitting based on subspace perturbation analysis with applications to structure from motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):841–854, 2009. 1, 2

[16] F. Jiang, M. Oskarsson, and K. Astrom. On the minimal problems of low-rank matrix factorization. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, June 2015. 2

[17] R. Kennedy, L. Balzano, S. J. Wright, and C. J. Taylor. Online algorithms for factorization-based structure from motion. *Computer Vision and Image Understanding*, September 2016. 1, 2

[18] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 1, pages I–1018–I–1025, 2001. 1, 2

[19] D. Martinec and T. Pajdla. 3d reconstruction by fitting low-rank matrices with missing data. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 1, pages 198–205, 2005. 1, 2

[20] B. Nasihatkon, R. Hartley, and J. Trumpf. A generalized projective reconstruction theorem and depth constraints for projective factorization. *International Journal of Computer Vision*, pages 1–28, 2015. 2, 3, 4

[21] D. Nistér, F. Kahl, and H. Stewénius. Structure from motion with missing data is np-hard. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–7, 2007. 2

[22] J. Oliensis and R. Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2217–2233, Dec 2007. 1, 2

[23] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *Image Analysis*, pages 524–535. 2011. 7, 8

[24] M. Oskarsson, K. Batstone, and K. Astrom. Trust no one: Low rank matrix factorization using hierarchical ransac. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 8

[25] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 4, 6, 7, 8

[26] P. F. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision (ECCV)*, pages 709–720, 1996. 1, 2, 4

[27] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello. Hierarchical structure-and-motion recovery from uncalibrated images. *Computer Vision and Image Understanding*, 140, November 2015. 4, 8

[28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 1, 2

[29] P. Torr and A. Zisserman. Mlesac. *Computer Vision and Image Understanding*, 78(1):138–156, Apr. 2000. 5

[30] B. Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 845–851, Jun 1996. 2, 4

[31] T. Ueshiba and F. Tomita. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In *European Conference on Computer Vision (ECCV)*, volume 1406 of *Lecture Notes in Computer Science*, pages 296–310, 1998. 1, 2