Revisiting Projective Structure for Motion: A Robust and Efficient Incremental Solution

Ludovic Magerand and Alessio Del Bue, *Member, IEEE*

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. To prevent error accumulation, a refinement based on alternation between new estimations of views and points is used. This can also be done with constrained non-linear optimization. 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%).

Index Terms—Structure-from-Motion, perspective cameras, projective reconstruction

1 Introduction

1.1 The Projective Structure from Motion Problem

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 [1], [2]. However, the Projective Structure from Motion (PSfM) [3] problem still entails difficulties and despite considerable efforts, there are clear limitations in current approaches [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [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 [15] 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 results in image projections that also depend non-linearly on the 3D points depths with respect to the camera, giving a slightly different problem:

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

where $\mathbf{1}_3$ is a 3-vector of ones and $\odot$ or $\otimes$ denote respectively the Hadamard or Kronecker products. The matrix D of size $f \times n$ contains coefficients named *projective depths* which are related to the real depth of every points in each camera frame.

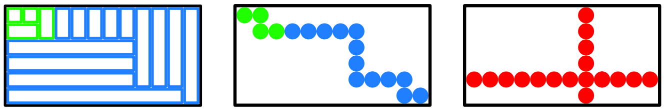
Moreover, when dealing with images having wide baselines, it is rather common to have 3D points occluded either by the scene itself or because being out of the camera field. As a consequence, the matrix M is often incomplete with some of its entries missing. Completing these entries leads to an NP-hard problem [16], [17] that can be defined as:

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

where Z is a $f \times n$ binary matrix corresponding to the set $\mathcal{Z}$ of known entries.

- *L. Magerand is now with the Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) at the Czech Technical University (CTU) in Prague. He was previously working on this topic at the Visual Geometry and Modelling (VGM) Lab of the Istituto Italiano di Tecnologia (IIT).*
- *A. Del Bue is with the Visual Geometry and Modelling (VGM) Lab of the Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy.*

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 is the reason why previous methods were able to solve mainly toy-problems. Here we provide a practical and incremental method, named P^{2}SfM, that allows to solve for projective reconstructions with dataset that represents a challenge even for modern large-scale solvers.



(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.

Fig. 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.

1.2 Notation

Homogeneous coordinates of a vector $\mathbf{v}$ are written as $\tilde{\mathbf{v}} = [\mathbf{v}^\top 1]^\top$ and $\mathsf{I}_{m\times n}$ is the $m \times n$ identity matrix. A $m \times n$ matrix of 0 (or 1) is denoted $\mathsf{0}_{m\times n}$ (or $\mathsf{1}_{m\times n}$) 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 $[\mathbf{v}]_\times$.

1.3 Related Work

Tomasi and Kanade [15] 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 [3], [18] 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, [4] showed that this method can sometime converge to useless results.

There have been several attempts to improve Sturm and Triggs solution [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] 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 $\mathsf{D} = \mathsf{1}_{f\times n}$ 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. An exception is [12] which proposes a SDP formulation based on a trace norm minimization making it suitable for global optimization. Only few of the mentioned methods [10], [11], [12], [13] 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.* [19] 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. In particular, the GPRT 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. 1c. Except for [12], the previous methods mentioned above do not comply fully with this theorem as they usually do not prevent the cross-shaped configuration. In [19], two set of constraints were proposed to ensure compatibility with the GPRT. The first one is linear and fixes some entries of the matrix D following a step-like mask as in Fig. 1b. The second one is quadratic and considers all entries of the matrix D by considering the norm of non-overlapping tiles such as in Fig. 1a.

Another important aspect for a practical factorization method is its ability to deal with missing and erroneous data. In particular for SfM, if we exclude toy problems, the measurement matrix M has a large number of missing data given that few views have many overlapping points. Solutions to solve this problem started with the method of Wiberg [20], [21]. A set of methods propose strategies that combine partial low-rank factorizations from sub-blocks of the M matrix that contain full data only. Although applied mainly to affine camera models, the optimization strategy used is quite relevant. This technique was pioneered by Jacobs [22] reconstructing the measurement matrix by first building its row or column null-space or one of its range spaces and have been applied both to the rigid [22] and non-rigid [23] SfM problems. This part based factorization has been then extended to be more robust to higher ratios of missing data and outliers [1], [2], [24], [25], [26]. However these solutions, more computationally viable, are 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 [2] limited to Henneberg constructive extensions and adapted to the perspective case.

In the case of affine SfM, the use of the L1-norm has been proposed to handle missing data and outliers [27], [28]. Large scale matrix factorization with noise is also a topic of interest in recommendation systems where the non-negative constraint makes the problem harder. To deal with this and given large size matrices, online and incremental approaches have been proposed [29], [30] and associated to gradient descent [31], [32]. Finally, the Divide and Conquer approach where the matrix is divided into small blocks has been used successfully to solve this kind of problem [33].

1.4 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 parametrization** of the PSfM problem which is still bilinear. Moreover, we show that a generalization of the step-like mask constraint on the projective depths of [19] 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 [2], 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 [34]. Whenever the reconstruction obtained is incomplete, *i.e.* does not contain all views, we propose to restart it from unestimated views and merge the reconstructions if possible.

2 Compact Factorization 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 [35] 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 [35], 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) giving

$$\mathsf{E}\left(\tilde{\mathbf{m}}_{i,j}\tilde{\mathbf{m}}_{i,j}^{+} - \mathtt{I}_3\right)\mathsf{X}_i\mathbf{q}_j = \mathbf{0}. \tag{6}$$

Again the matrix E is used to remove redundancy as $\tilde{\mathbf{m}}_{i,j}\tilde{\mathbf{m}}_{i,j}^{+} - \mathtt{I}_3$ is rank deficient. While this provides a maximum likelihood estimator, it seems less accurate than the DLT elimination as will be shown experimentally in Sec. 4.1.4.

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

$$\begin{bmatrix} \mathsf{E}\,\mathsf{F}_{1,j}\,\mathsf{X}_1 \\ \vdots \\ \mathsf{E}\,\mathsf{F}_{v,j}\,\mathsf{X}_v \end{bmatrix}\mathbf{q}_j = \mathbf{0}_{2v}, \tag{7}$$

where $\mathsf{F}_{i,j}$ is the matrix $\left[\tilde{\mathbf{m}}_{i,j}\right]_{\times}$ or $\tilde{\mathbf{m}}_{i,j}\tilde{\mathbf{m}}_{i,j}^{+} - \mathtt{I}_3$ depending on the elimination method chosen.

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

$$\begin{bmatrix} \mathsf{E}\,\mathsf{F}_{i,1}\,\mathsf{G}_1 \\ \vdots \\ \mathsf{E}\,\mathsf{F}_{i,p}\,\mathsf{G}_p \end{bmatrix}\mathbf{x}_i = \mathbf{0}_{2p}\ ,\ \mathsf{G}_j{}^{\top} = \begin{bmatrix} \mathbf{q}_j & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{q}_j & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{q}_j \end{bmatrix}. \tag{8}$$

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

$$\underbrace{\left(\frac{1}{k_j^p}\sum_{i\in\mathcal{F}_j^p}\tilde{\mathbf{m}}_{i,j}^{+}\mathsf{X}_i\right)}_{=\mathbf{c}_j^{p\top}}\mathbf{q}_j = 1 \ \text{ or } \ \underbrace{\left(\frac{1}{k_i^v}\sum_{j\in\mathcal{F}_i^v}\tilde{\mathbf{m}}_{i,j}^{+}\mathsf{G}_j\right)}_{=\mathbf{c}_i^{v\top}}\mathbf{x}_i = 1, \tag{9}$$

where $\mathcal{F}_j^p$ is the set of k_j^p views used to constrain the point j and $\mathcal{F}_i^v$ the set of k_i^v points used to constrain the view i. Note that the measurements must be available for all the projections considered into these sets. However we do not have to necessarily consider all the visible projections as explained in App. A.

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 [19] and illustrated in Fig. 1b, which corresponds to the tiling of Fig. 1a. 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. 3.3 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.

2.3 Solving for Projective Parameters

The systems from Eq. (7) and Eq. (8) are homogeneous and linear in either $\mathbf{q}_j$ or $\mathbf{x}_i$. They can be written generically as

$$\mathsf{A}\mathbf{y} = \mathbf{0}, \tag{10}$$

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

The constraints from Sec. 2.2 can then be used to linearly substitute one of the projective parameters in these systems. Doing the substitution to remove y_1, the first entry of $\mathbf{y}$ in Eq. (10), we have to split $\mathbf{A}$, $\mathbf{y}$ and $\mathbf{c}$ defined in Eq. (9) as

$$\mathsf{A} = \begin{bmatrix}\mathbf{a} & \mathsf{A}'\end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \mathbf{z}\end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \mathbf{c}'\end{bmatrix}, \tag{11}$$

where $\mathbf{a}$ and A' are the first and remaining columns of A. Then after the substitution, we obtain the following standard linear system satisfied by $\mathbf{z}$

$$\mathsf{B}\mathbf{z} = \mathbf{b} \qquad \text{with} \quad \left\{ \begin{array}{l} \mathsf{B} = \mathsf{A}' - \mathbf{a}{\mathbf{c}'}^\top / c_1 \\ \mathbf{b} = -\mathbf{a}/c_1 \end{array} \right. . \tag{12}$$

With at least two visible projections for a point and six for a view, this system is overdetermined and can be efficiently minimized in the Least Squares sense to solve for $\mathbf{z}$ using an economy size QR decomposition with column pivoting of the matrix B and back substitution in the triangular factor. Note that $\mathbf{z}$ is a minimal parametrization of the projective parameters as it contains only three degrees of freedom for a point and eleven for a view. Once $\mathbf{z}$ is estimated, we can retrieve y_1 as

$$y_1 = \frac{1}{c_1}\left(1 - {\mathbf{c}'}^\top \mathbf{z}\right), \tag{13}$$

which gives a minimizer $\mathbf{y}$ of Eq. (10) satisfying exactly the constraint $\mathbf{c}^\top \mathbf{y} = 1$ from Eq. (9).

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

We describe here our method to estimate the projective factors X and Q by solving incrementally and robustly

$$\min_{\mathsf{X},\mathsf{Q}} \sum_{i,j \in \mathcal{Z}} \|\mathsf{EF}_{i,j} \mathsf{X}_i \mathbf{q}_j\|_2^2 \qquad s.t. \quad \left\{ \begin{array}{l} {\mathbf{c}_j^p}^\top \mathbf{q}_j = 1, \ \forall j \\ {\mathbf{c}_i^v}^\top \mathbf{x}_i = 1, \ \forall i \end{array} \right. , \tag{14}$$

which minimizes the algebraic error in the projective space under constraints complying with the GPRT [19] to ensure a valid reconstruction. A graphical illustration of our approach is provided in Fig. 2 and important details on each step are given from Sec. 3.3 to 3.6. We named our method *Practical Projective Structure-from-Motion* and abbreviated it P^{2}SfM to avoid spelling difficulties.

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 (Sec. 3.2). It is more computationally efficient to compute all $\tilde{\mathbf{m}}_{i,j}^{+}$ and $\mathsf{EF}_{i,j}$ only once and store them into sparse data matrices. The method then starts with an initial sub-problem (Sec. 3.3) and iterates by robustly adding missing tiles (Sec. 3.5) 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. 3.4). After each inclusion, the reconstruction is refined by re-estimating all the points and views already added (Sec. 3.6). The complete method is then given in Fig. 3 and can be restarted from line 3 to provide multiple reconstructions when the first reconstruction does not contain all views. The result is multiple normalized projective reconstructions satisfying the GPRT [19] and the corresponding sets of inlier projections.

3.2 Normalization of Projections

At the very beginning of our method, line 1 of Fig. 3, we normalize the measurement matrix before any other operations. Although done following the standard procedure of [35], this step is of crucial importance as the linear Least Squares optimization done later can become unstable without it. For each view of the problem, we compute and apply a 3×3 transformation given by a 2D translation moving the centroid of the measurements to the center of the image and an isotropic rescaling such that the average Euclidean distance to the center is $\sqrt{2}$. This is done only for the visible tracked points in this view and the computed transformation is stored to be reused later for the computation of the reprojection error. Note that the result of our method is therefore a normalized projective reconstruction which can be rectified to match the original image frames by inverting the transformations. It is however preferable to make this last step after the metric upgrade as it can also benefit of the normalization.

3.3 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. 2a that can be robustly solved to get a valid initial projective reconstruction. This is done in the standard way with robust fundamental matrix estimation [35] after selecting two views using the pyramidal affinity score described in App. B. 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. When restarting to provide multiple reconstructions, we consider only pairs of views containing at least one view that has not been estimated in any of the previous reconstructions. We also prefer pairs which contain only views not previously estimated.

After extracting the epipolar geometry from the estimated fundamental matrix as in [3], [18], an SVD of the submatrix 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. This balancing procedure is detailed in App. A.

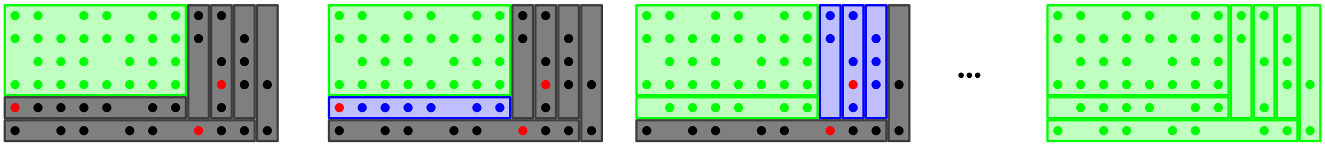

(a) Initial sub-problem (green). (b) Adding one view (blue). (c) Adding three points (blue). (d) Final reconstruction and inliers.

Fig. 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).

```
1  Normalize projections, see sec. 3.2, and compute all
     data matrices ;
2  while some views are not estimated and maximum
     number of model not reached do
3      Find an initial sub-problem and robustly solve it,
         see sec. 3.3 ;
4      while reconstruction is not complete and
         (reconstruction was extended or an eligibility
         threshold can be decreased) do
5          Find currently eligibles views, see sec. 3.4 ;
6          Try to add eligibles views robustly, see
             sec. 3.5 ;
7          if at least one eligible view has been added then
8              Increase the eligibility threshold for
                 points ;
9              Refine locally the reconstruction, see
                 sec. 3.6 ;
10         else if no view was eligible then
11             Decrease the eligibility thresholds for
                 views if not minimum ;
12         Find currently eligibles points, see sec. 3.4 ;
13         Try to add eligibles points robustly, see
             sec. 3.5 ;
14         if at least one eligible point has been added then
15             Increase the eligibilty thresholds for
                 views ;
16             Refine locally the reconstruction, see
                 sec. 3.6 ;
17         else if no point was eligible then
18             Decrease the eligibility threshold for
                 points if not minimum ;
19     Refine globally the reconstruction, see sec. 3.6 ;
```

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

3.4 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 [34] of the visible known points and select only the views with highest scores.

If the eligibility thresholds are set too high, 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 between an initial high value and a minimum value both provided by the user. The thresholds are decreased only when we are unsuccessful in extending the reconstruction (lines 11 and 18 of Fig. 3) and increased again when we successfully extends it (lines 8 and 15 of Fig. 3). The idea behind this dynamic approach is to maximize the quantity of data used to estimate new points and views, which improves the quality of the reconstruction.

We also included a rejection mechanism for points or views that 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.

3.5 Robustly Adding a Point or View

Our method is based on minimizing the linear Least Squares systems of Eq. (12) 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 [36] with the MSAC score [37] 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. (12). The two first cases can happen with degenerate configurations of points or views but more frequently when estimating a view [38]. In order to get better estimation from the random subset, we also increased slightly 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.

3.6 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 reconstruction after trying to add eligible points (or views) if any addition was successful. This 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 [8] but without any constraints on the projective depths.

To speed up the process while doing the completion, the refinement in line 9 and 16 of Fig. 3 is done locally, meaning only over the views and points recently added. By recently, we intend only those added after the two last changes of direction in the reconstruction extension which corresponds to switching from adding only views to adding only points (or the other way around). Note that it does not necessarily coincide with the two last iterations of the main loop as this loop can add only views (or only points) in many consecutive iterations.

The refinement in line 19 of Fig. 3 is done either locally or globally after more than a fixed number of local refinement has been done. When it is done globally, all the estimated points and views are reestimated. This is to ensure that the improvement made locally is propagated to the entire reconstruction. When the main loop ends, a final and global refinement is also done.

Finally, the refinement can also be done by minimizing directly the problem given in Eq. (14), which should be preferable as all parameters would be optimized at once (*i.e.* without alternation). This can be achieved using the current estimation as an initialization for a local non-linear optimization based on the augmented Lagrangian method coupled with the L-BFGS algorithm [39]. To be efficient, this requires analytical computation of all gradients, which are easily done as both objective and constraints are polynomial functions in the unknowns. Providing analytical Hessian of the Lagrangian also allows for faster and more accurate steps. All these computations have to be parallelized in order to achieve maximum efficiency. Note that although the problem of Eq. (14) is polynomial with a low degree, the scale of it prevents the use of polynomial optimization methods.

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. Our implementation is freely available online[1]. We compared our method (P^{2}SfM) with [13] (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 [40]. 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. 4a. This impacts also the 3D points reconstruction error which grow quickly for them in Fig. 4b. Differently, our approach shows strong resilience to increasing number of outliers.

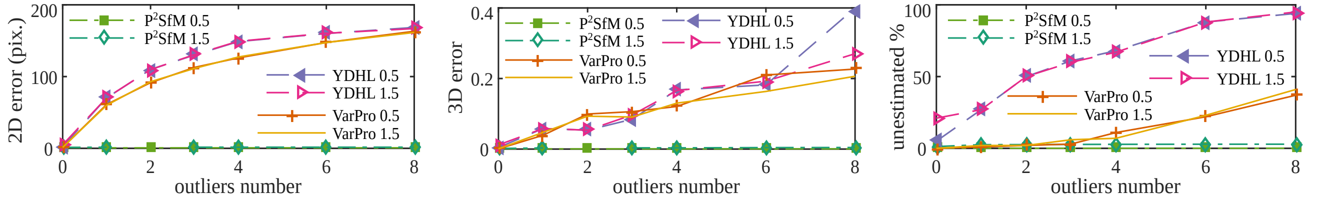
Note that for a given sequence, previous works return a result for all points and views or for none of them while our method always gives a reconstruction where some points or views might be unestimated due to the rejection mechanism of Sec. 3.5. In Fig. 4c, the entire synthetic dataset is considered 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. 5a 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.

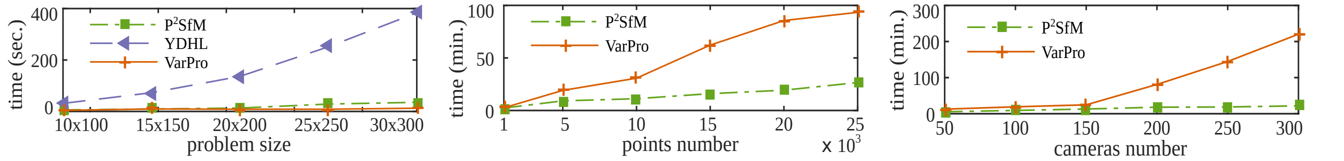
When dealing with medium scale sequences, Fig. 5b(c) show the behaviour when increasing the number of points

[1] Website is at https://bitbucket.org/lmagerand/ppsfm



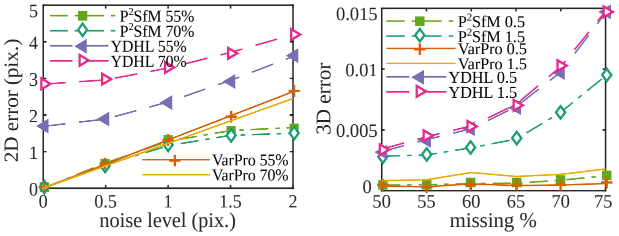
(a) 2D reprojection error. (b) 3D points error. (c) Unestimated points.

Fig. 4. 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.

Fig. 5. 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.

Fig. 6. 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.

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 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. 6a 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. 6b 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.

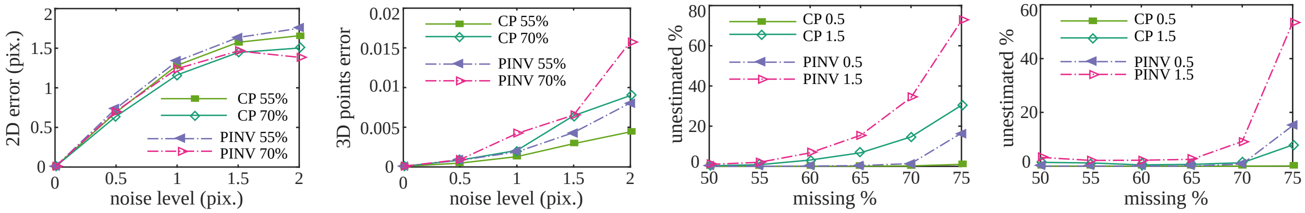
4.1.4 Comparison of Elimination Methods

Using the same dataset as in Sec. 4.1, we compared the elimination of the projective depths done in Sec. 2.1 using the cross product and the pseudo inversion. By varying noise level from 0 to 2 pixel and missing data ratio from 50% to 75%, we can see in Fig. 7 that the behaviour is globally similar. The cross product elimination achieves a lower average 3D points and reprojection error as visible in Fig. 7a and 7b. This is probably caused by the robust elimination and rejection mechanism activated more often resulting in less data available left to make the estimation as confirmed in Fig. 7c and 7d which display the number of unestimated points and views.

4.1.5 Non-Linear Refinement

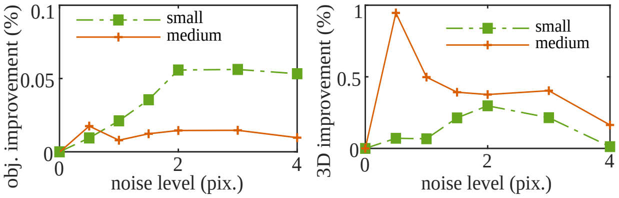
To evaluate the non-linear refinement described in Sec. 3.6, we tested it as a complement to the finale refinement done in line 19 of Fig. 3 with the same small scale dataset as in Sec. 4.1. A larger scale synthetic dataset made of twenty sequence with 100 views and 2000 points was also generated. The implementation was done using the `fmincon` function of MATLAB with user-supplied gradients for both the objective function and constraints without parallelization. The Hessian of the Lagrangian was not provided.

As it can be observed in Fig. 8a and 8b, the improvement on both the cost function and the 3D points error is lower than 1% whatever the noise level added to the measurements. This means that the alternation based refinement scheme is already providing good reconstructions and the non-linear refinement does not improve consistently. Moreover, at 2 pixels noise level, it takes 25 iterations on



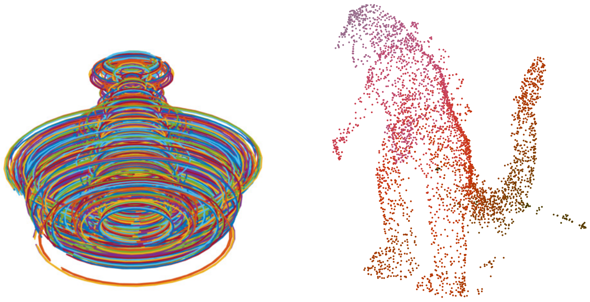
(a) Reprojection error. (b) 3D points error. (c) Unestimated points. (d) Unestimated cameras.

Fig. 7. Comparison of the cross product (CP) and pseudo inversion (PINV) eliminations. (a) and (b) are respectively the 2D reprojection error and 3D structure error with increasing noise level at two ratio of missing data (55% and 70 %). (c) and (d) are respectively the percentage of unestimated points or views when the missing data ratio is increasing at two level of noise ($\sigma = 0.5$ and $\sigma = 1.5$ pixels).



(a) Cost function. (b) 3D points error.

Fig. 8. Improvement made by the non-linear refinement on the objective function (a) and 3D points error (b) compared to the output of the alternation based refinement. The gain in accuracy is minimal for both the small scale dataset and the larger one.



(a) Completed measurements. (b) 3D reconstruction.

Fig. 9. 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.

average to reach the minimum for both dataset leading to excessive computation time. While computing the Hessian analytically could eventually improve this, the small gain of accuracy make the implementation effort questionable.

4.2 Real Data

In Tab. 1, we also evaluated P^{2}SfM on various real datasets of different size available in the literature. Note that, following COLMAP [34], points track with only two visible projections have been removed from M in medium and large scale sequences. These points are unreliable and useless for expanding the reconstruction further and this also speed-up the search for eligible views and points as the size of the visibility matrix is reduced. The reported size for each sequence is given after this removal.

When necessary, feature extraction and matching have been done off-line once for all methods prior to reconstructions using the first stage of COLMAP [34] and we built the measurement matrix from the output using [44] to find points tracks. To obtain an Euclidean reconstruction from the projective one, we used the metric upgrade method of [45]. Given timings do not include the time required for all these steps.

4.2.1 Small Scale Sequences

Five small scale sequences containing less than a million entries in the measurements matrix M were evaluated and results are given in Tab. 1. As points tracks are usually shorter in small scale sequences, points with only two visible projections are very common. Therefor, we have set the minimum eligibility threshold to two visible projections for these experiments only. 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. 9b 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. 9a. We used the 4983 points experiment since the smaller Dino sequence is mostly suited for affine structure from motion approaches [46] and it has a low missing data ratio. The non-linear refinement was tested on this sequence, and a 3% improvement is obtained on the objective function. As it takes more than 4 minutes of computation even on this small sequence, the non-linear refinement was not tested on any other real data.

4.2.2 Medium and Large Scale Sequences

As seen in Tab. 1, 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 any of these sequence 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 [34], a standard bundle adjustment based method implemented in C++ using highly optimized libraries and a camera model with radial distortion. While COLMAP usually achieves a lower reprojection error, we need much less computational time for the same reconstruction size. Example of views and

TABLE 1
Real Sequences Results. P^{2}SfM, VarPro and YDHL were evaluated on five 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 seven sequences against VarPro and COLMAP. VarPro could not provide a result for these sequences while COLMAP is usually slower than P^{2}SfM for approximately the same reconstruction size. The last sequence is an example of multi-model reconstruction which are merged after the metric upgrade.

Sequence			P^{2}SfM			VarPro [14]		YDHL [13] or COLMAP [34]		
Name	Size	Missing	Reconstructed	2D error	Time (sec.)	2D error	Time (sec.)	2D error	Time (sec.)	
House (VGG)	10 × 672	57.7%	10 × 672	0.4284	2.38	0.6246	68.9	0.6639	1054	YDHL
Dinosaur 319	36 × 319	76.9%	36 × 319	0.4668	2.59	1.5761	90.8	3.3543	609	
Dinosaur 4983	36 × 4983	90.8%	36 × 4953	0.3780	11.6	1.6492	1428	Time Limit (4H)		
Wilshire (Ponce)	190 × 411	60.7%	190 × 411	0.4789	38.1	0.5995	3623	0.6688	3851	
Blue Teddy Bear (Ponce)	196 × 827	80.7%	196 × 827	0.6577	45.3	1.4169	13341	Time Limit (4H)		
Vercingetorix [41]	69 × 11743	95.39%	69 × 10835	0.3887	106	Time Limit (12H)		0.4220	92	COLMAP
Cherubim [42]	65 × 45153	93.3%	65 × 45002	0.9146	105	Time Limit (12H)		0.4827	182	
Dome des Invalides [41]	85 × 56031	91.9%	85 × 56031	0.3813	283	Out of Memory (8GB)		0.4115	278 [2]	
Arc de Triomphe [41]	173 × 35971	95.5%	173 × 35013	0.5292	276	Out of Memory (8GB)		Not Available (no images)		
Alcatraz Water Tower [41]	173 × 40515	94.5%	173 × 40228	0.7159	343	Out of Memory (8GB)		0.4226	1696	
Alcatraz West Side [41]	414 × 109214	96.6%	400 × 105064	0.8554	2442	Not Tested		0.5728	5141	
Orebro Castle [41]	763 × 146065	98.3%	763 × 142776	0.5989	4972	Not Tested		0.4953	28372 [3]	
Trafalgar Square [43]	257 × 25454	98.7%	222 × 13320	0.9760	689					

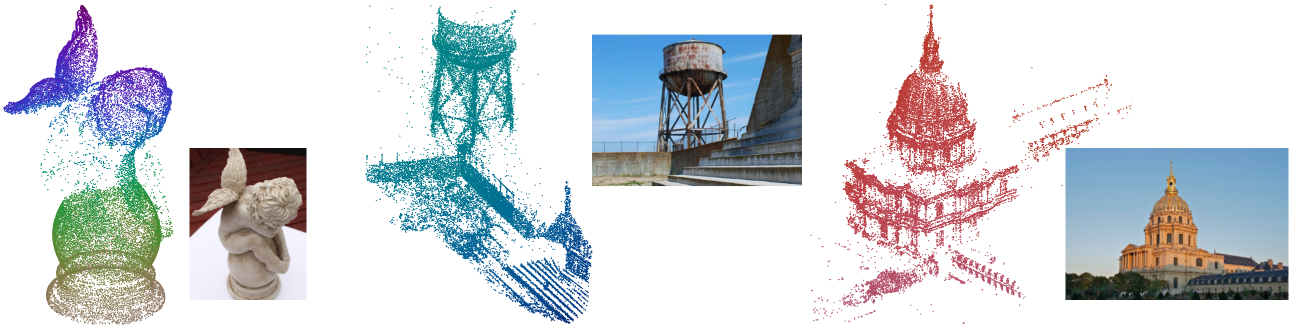

(a) Cherubim. (b) Alcatraz Water Tower. (c) Dome des Invalides.

Fig. 10. 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.

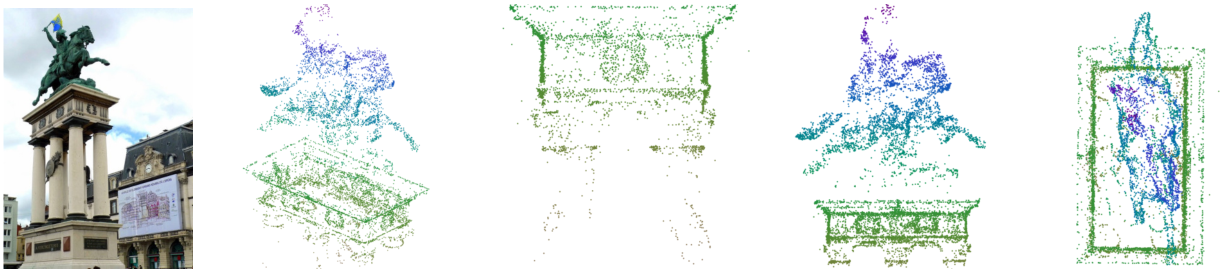

(a) Photo. (b) Overview. (c) Front. (d) Side. (e) Top.

Fig. 11. One photo of the Vercingetorix statue (not from the sequence) and various views of the reconstruction. Despite strong perspective effects due to the height of the statue, the reconstruction recovers correctly the shape with many details.

overview of the reconstruction are given in Fig. 10. Given the size of the Alcatraz West Side and Orebro Castle sequences (about 120 millions entries in M), to the best of our knowledge this are the largest successful reconstructions for a PSfM method. Detailed comments on each sequence are given next.

Vercingetorix. This sequence from [41] is made of close views taken around the statue of Vercingetorix in Clermont-Ferrand, implying strong perspective effects as the statue is several meters above the ground as seen in Fig. 11a. The statue itself is made of an horse carrying Vercingetorix and jumping over a soldier on the ground. Various body parts of the horse can be recognized in the reconstruction Fig. 11b: tail, legs and head. The shape of Vercingetorix itself atop the horse can be observed too, particularly in Fig. 11d. On the basis seen from the front in Fig. 11c and side in Fig. 11d, the ornaments and pillar heads are also reconstructed. The top view in Fig. 11e shows that the angles are corrects.

Cherubim. This sequence consists of high resolution images of a cherubim statue from [42]. It has a mostly uniform surface and colour in many parts, resulting into difficulties

[2] COLMAP reconstructed only 21108 points in this sequence.
[3] COLMAP full reconstruction has 249301 points. After 2 hours of computation, 365 views and 157821 points were reconstructed.

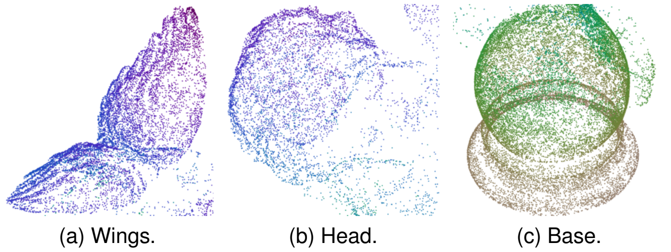

(a) Wings. (b) Head. (c) Base.

Fig. 12. Details of the cherubim statue reconstruction. The round shape of the base (c) is perfectly recovered. On the head (b), we can recognize some features of the face like eyes or nose. The reconstruction of the wings (a) is also highly detailed.

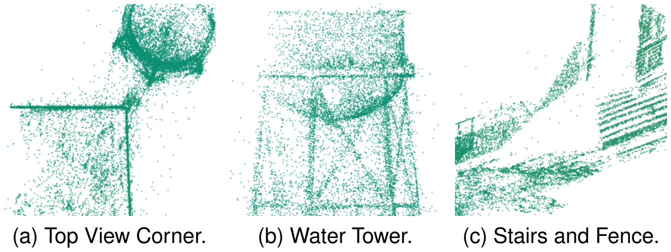

(a) Top View Corner. (b) Water Tower. (c) Stairs and Fence.

Fig. 14. Details of the reconstruction of the Alcatraz courtyard and water tower. An upper view of the corner of the courtyard (a) shows correct angle of the walls and circular shape of the water tower for which the structure is also nicely recovered (b). The stairs (c) are parallel and the first step is higher than the others as in the reality.

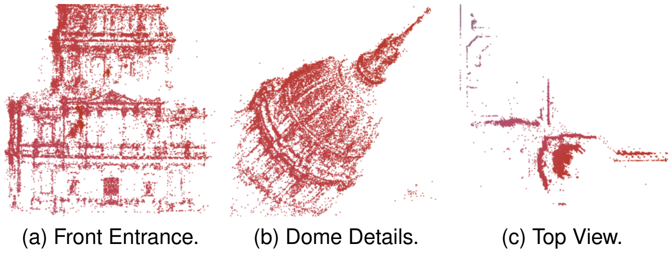
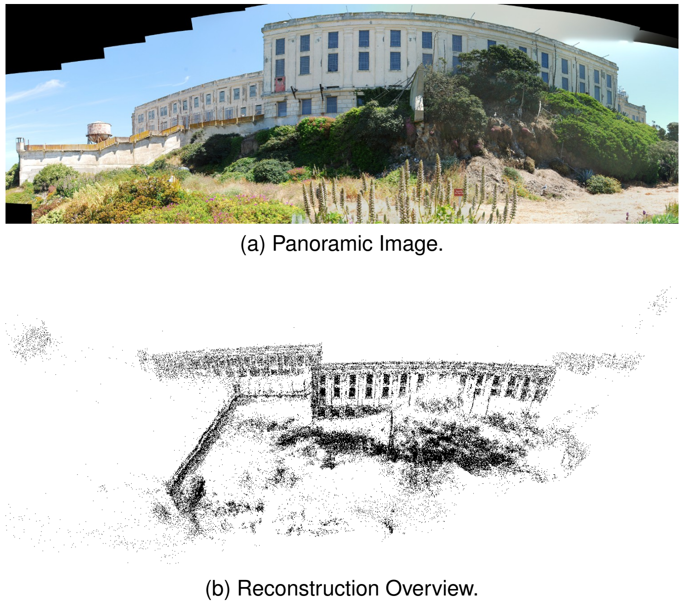

(a) Front Entrance. (b) Dome Details. (c) Top View.

Fig. 13. Details of the reconstruction of the Dome des Invalides. The front entrance (a) and the dome (b) of the main building. A global top view of the scene (c) including the garden, one of the wings and the rear buildings.


(a) Panoramic Image.

(b) Reconstruction Overview.

Fig. 15. Panoramic view built from images of the Alcatraz West Side sequence and the corresponding reconstruction of 105064 points obtained with P^{2}SfM in only 41 minutes. This is a reasonable reconstruction where all the main features of the scene are correctly recovered.

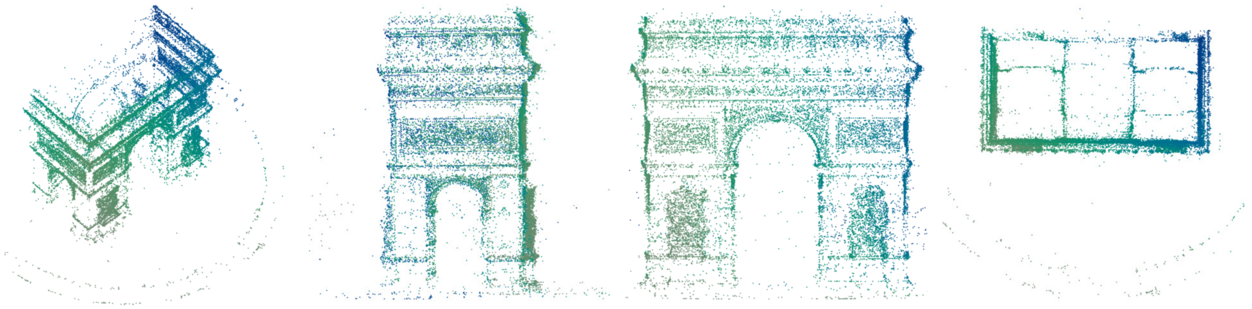
to detect and match feature points. However, where enough texture is present, the quality of the reconstruction and the details are remarkable as can be observed in Fig. 10a. The head is reconstructed with high details as it can be seen in Fig. 12b, the main features of the face are visible. Finally Fig. 12a shows that the shape of the wings is perfectly recovered as is the round shape of the base in Fig. 12c.

Domes des Invalides. This is another sequence of [41] which features the Domes des Invalides in Paris visible in Fig. 10c. A reconstruction of the dome itself is shown in Fig. 13b and it is well recovered, especially when considering that the building is tall and all the views are taken from the ground level. The top view of the reconstruction in Fig. 13c makes evident the correct angles of the reconstruction between the left wing and the rear building, even in the gardens where few points are tracked. In Fig. 13a, the details of the reconstruction go as far as providing the doors shape and statues in the alcoves. Note that COLMAP takes approximately the same time than P^{2}SfM to build a reconstruction that contains much less points (only 38%). This results from the method we use to find the points tracks [44] from the output of COLMAP matching which delivered more of them than COLMAP internal pipeline.

Arc de Triomphe. This sequence is made of images taken around the Arc de Triomphe in Paris, a monument 50 meters high with lot of detailed decorations. The proposed method recovered most of the fine details as well as the overall shape of the monument as it can be seen in Fig. 16. In Fig. 16d and Fig. 16a, the points circling around the monument delimit the roundabout where it is. The various decorations of the monument are clearly visible in Fig. 16b and Fig. 16c.

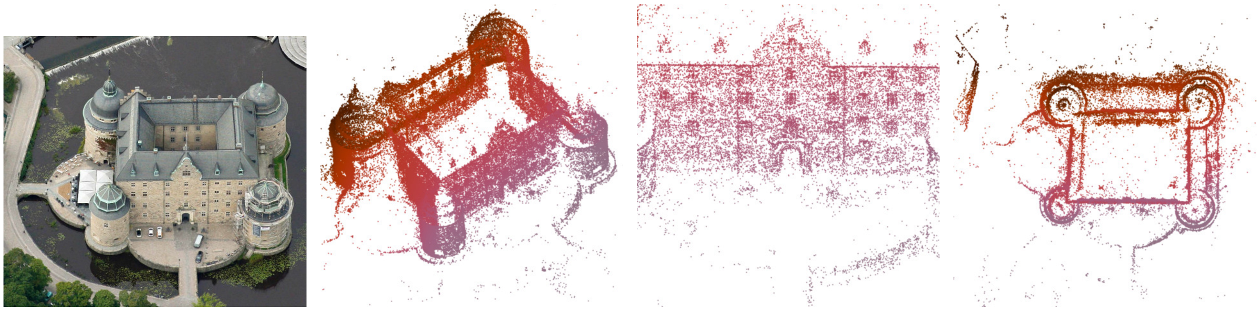
Alcatraz Water Tower. The scene displayed in this sequence is a corner of the courtyard in the Alcatraz prison with its water tower. This sequence is taken from [41] and contains lot of views where the perspective effect is clearly visible, especially on the stairs. A top view of the reconstruction is given in Fig. 14a to show how well the walls of the courtyard have been recovered. The cluster of points at the corner, outside the walls, comes from a small cage present in the scene although barely visible. The shape of the water tower is also correct, which is confirmed also in a view from its side as in Fig. 14b. Looking at the reconstruction from another angle, Fig. 14c shows the stairs where the first step is correctly reconstructed as being higher than the other.

Alcatraz West Side. As it can be seen in Fig. 15, this sequence shows the garden outside the main building of Alcatraz and it comes from [41]. The presence of an abundant vegetation, illumination variation and many similar features on the building made it relatively challenging. Nonetheless, the reconstruction obtained in Fig. 15b, is rendering all the main parts of the scene: the main building and its bushes, the wall between the garden and the courtyard including its fence, the top of the water tower and the lighthouse.



(a) Overview. (b) Overview. (c) Front. (d) Top.

Fig. 16. Details of the reconstruction of the Arc de Triomphe. The rectangular shape of the monument is well recovered as can be seen in (c) and (c). The sculptures and reliefs are also clearly distinguishable in (a) as well as the circular barrier around the monument.



(a) Aerial Photo. (b) Overview. (c) Entrance. (d) Top View.

Fig. 17. Details of the reconstruction of the Orebro castle. The overview (b) presents it from an upper view above the corner corresponding to the smallest tower. The main entrance (c) of the castle can be seen on the front wall. Notice that in the top view (c) some walls appear to not be orthogonal. This effect is not a perspective distortion given by our method since walls are not orthogonal in the real world.

Notice that the angle of the wall between the garden and the courtyard might seem incorrect but is actually like this in reality. Compared to COLMAP, we are twice faster to achieve a reconstruction similar in size.

Orebro Castle. The last sequence from [41] shows the castle of Orebro for which an aerial photo can be seen in Fig. 17a. The shape of the castle is perfectly recovered as can be observed in Fig. 17b, including many details from the front wall in Fig. 17c such as the entrance and windows. Every tower has a different size and this is also visible in the reconstruction from the top view in Fig. 17d. This is the largest sequence we reconstructed with our method and it takes less than an hour and half. Although COLMAP reconstructs 1.7 times more points, it also takes 5.7 times much more time. As previously, the difference comes from the method used to build the point tracks. This is independent of our method which still reconstructs more than 97% of the points tracks available.

4.3 Multiple Model Reconstruction

The famous Trafalgar sequence from [47] was one of the first successful large scale reconstruction done from 257 internet photos taken around the Trafalgar square in London. This sequence is very challenging as it contains a lot of very noisy measurements due to the various light variations and scene change between all photos. It is also an extremely sparse dataset as the missing data ratio is of 98.65% and only 39% of the points are visible in more than two views. The visibility matrix given in Fig. 18 clearly shows two blocks and few common data. Therefore, the core of our method (*i.e.* line 3 to 19 of Fig. 3) is unable to go from one block to the other most of the time as not enough of the common data pass both outliers rejection and eligibility thresholds. By decreasing the later and increasing the former, it is possible to obtain a single model reconstruction but with a relatively low quality as can be observed in Fig. 19c especially on the buildings in front of the National Gallery.

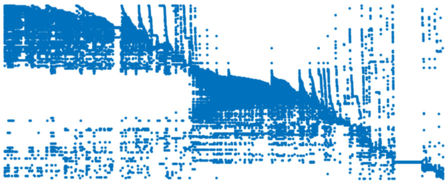


Fig. 18. Visibility matrix of the Trafalgar square sequence where two blocks of data with few common parts can be isolated.

However as seen in Fig. 19a, keeping safer values and allowing for multiple models by restarting the main method in line 2 of Fig. 3 enable us to obtain a better reconstruction of both blocks which can be merged after the metric upgrade using a robust procrustes analysis based on RANSAC. The reconstruction and merging of the two blocks has been done in 12 minutes, the first block contains 107 views and 6445 points while the second one is made of 134 views and 7314 points. There are 19 views and 439 points common to both

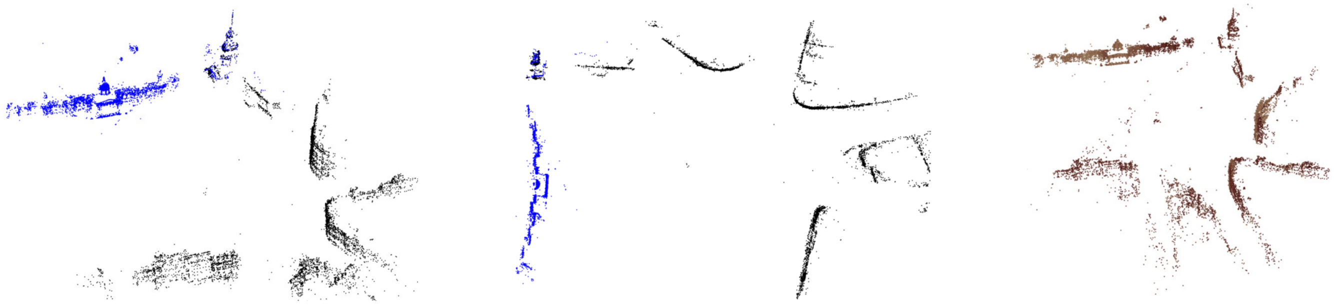

(a) Overview. (b) Top View. (c) Single Model.

Fig. 19. Reconstruction of the Trafalgar sequence obtained by merging the two models resulting of our method. The first block (in blue) contains the National Gallery and Saint Martin church while the second one (in black) is made of the Saint Martin church and some of the buildings around the square. A single model reconstruction such as (c) can be obtained by lowering the settings which result in a poorer quality compared to (a).

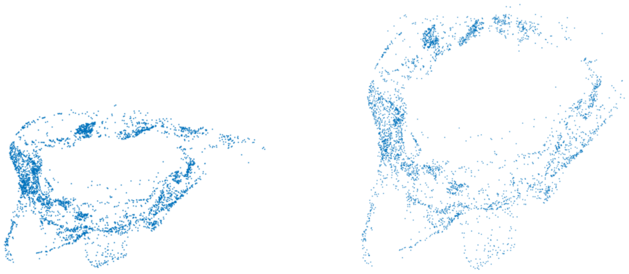

(a) Incorrect. (b) Correct.

Fig. 20. Example of the loop-closure problem in the reconstruction of the Linnaeus statue.

blocks used for the merging giving the final reconstruction of 222 views and 13320 points. We tried allowing more than two models but the next ones are too small and cannot be merged with the two main models as they do not have enough common points.

5 Discussion

Despite the precautions adopted in Sec. 3.5, it can happen that degenerate configuration occur. Sometime many attempts are required before obtaining a successful reconstruction depending on the path taken while adding views and points. Preventing this from happening would require a set of reliable geometric verifications after the estimation of a new view (or point) which are not currently implemented.

In the Linnaeus sequence from [2], our method fails to achieve a correct reconstruction most of the time. As it can be observed in Fig. 20a compared to a correct reconstruction in Fig. 20b, an effect similar to the loop closure problem is present where the reconstruction ends. We believe that in this sequence, the current refinement scheme based on alternation might not be enough to prevent error accumulation.

5.1 Implementation Details

In order to obtain best results, we recommend setting the minimum eligibility thresholds to at least 24 for views and 4 for points whenever possible, and strongly advise to not set them below 18 and 3 respectively. We used initials values of 48 for views and 6 for points. For most sequences, the parameters for the robust estimation were: an outlier threshold of 4 pixels, a maximum number of iterations fixed at 2000, 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^{-4} or 50 iterations have been done. The final refinement is made with a threshold of 10^{-5} over the parameters change or a maximum of 100 iterations.

6 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 [2], [48] so allowing to reconstruct even larger sequences and scaling up our method. 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.

Appendix A
Balancing Projective Parameters

The GPRT [19] states that D from Eq. (1) should be diagonally equivalent to the matrix Λ of the real depths of each point with respect to each camera. This means that there exists two diagonal matrices (L, R) of size $f \times f$ and $n \times n$ such that

$$\mathsf{L}\ \mathsf{D}\ \mathsf{R} = \Lambda. \tag{15}$$

Moreover, to prevent wrong reconstructions, the theorem implies that these matrices must satisfy two conditions: The diagonal coefficients must not be null or lead to a cross-shaped D as in Fig. 1c. They are otherwise free and the goal of the constraints is to fix them so that the optimization does not converge to the trivial solutions where they would be null. As we have stated in Sec. 2.2, constraints on the projective depths can be transferred to the projective parameters.

The diagonal coefficients matrices can also be transferred to the projective parameters giving

$$\bar{\mathsf{P}}\,\bar{\mathsf{S}} = (\mathsf{L}^{-1} \otimes \mathtt{I}_3)\,\mathsf{X}\,\mathsf{Q}\,\mathsf{R}^{-1}, \tag{16}$$

where $(\bar{\mathsf{P}}, \bar{\mathsf{S}})$ are the real cameras and points matrices.

When solving the initial sub-problem in Sec. 3.3, we use a fundamental matrix estimation to compute the initial projective parameters. As the proof of the GPRT is based on considerations over the fundamental matrix, these initial projective parameters are naturally a valid projective reconstruction. However, they do not satisfies the constraints we defined in Sec. 2.2 but, thanks to the freedom of the diagonal coefficients, they can be rectified. In order to do so, we rescale them by α_j or β_i defined as

$$\frac{1}{\alpha_j} = k_j^p \left(\sum_{i \in \mathcal{F}_j^p} \tilde{\mathbf{m}}_{i,j}^{+} \mathsf{X}_i \right) \mathbf{q}_j \text{ or } \frac{1}{\beta_i} = k_i^v \left(\sum_{j \in \mathcal{F}_i^v} \tilde{\mathbf{m}}_{i,j}^{+} \mathsf{G}_j \right) \mathbf{x}_i \tag{17}$$

respectively for point $\mathbf{q}_j$ or camera X_i. This has no effect on the validity of the reconstruction as this transformation is absorbed into the diagonal coefficients.

For the same reason, we can consider only some entries of each tile in Eq. (9) by removing some projections in the sets $\mathcal{F}_j^p$ and $\mathcal{F}_i^v$. This just results in applying a rescaling α'_j or β'_i computed as in Eq. (17) except that the sum would be done over the $k_j^{p\prime}$ or $k_i^{v\prime}$ projections still considered.

Appendix B
Pyramidal Affinity Score

The selection of the initial sub-problem is important to the quality of the final reconstruction. To address this problem, we merge two ideas: the pyramidal visibility score defined in COLMAP [34] and the affinity score of Samantha [48]. Both are bundle adjustment pipelines focused on high quality reconstructions and error containment, the former working in a online manner and the later with a hierarchical procedure.

In Samantha, in order to build the hierarchy of actions to follow in the reconstruction, an affinity score between two images (i, j) is defined as

$$a_{i,j} = \frac{1}{2}\frac{|\mathcal{S}_{i,j} \cap \mathcal{S}_{j,i}|}{|\mathcal{S}_{i,j} \cup \mathcal{S}_{j,i}|} + \frac{1}{2}\frac{ch(\mathcal{S}_{i,j}) + ch(\mathcal{S}_{j,i})}{A_i + A_j}, \tag{18}$$

where $\mathcal{S}_{k,l}$ is the set of visible projections in image k also visible in image l, $ch(\mathcal{S}_{k,l})$ is the area of the convex hull of this set in image k and A_k is the total area of image k. The first term of this affinity score can be seen as a quality measurement of the match between the two views: the more common points they have, the better. The second term is a measurement of the scene overlap of the two views which is important to obtain a good stereo reconstruction.

In COLMAP, when looking for the next view to add to the reconstruction, a score based on the distribution of the projections of the already reconstructed points in the image is defined. This score uses an efficient multi-resolution analysis of the projections positions in order to maximize the quality of the camera resection to be done. At each layer l of the multi-resolution analysis, the image is divided into $2^l \times 2^l$ cells to which is associated a binary value: true if the cell contains a visible projection, false otherwise. Then the final score is

$$p = \sum_{l=1}^{n} (2^l)^2 c_l, \tag{19}$$

where c_l is the number of non-empty cells in the layer l.

To select the initial sub-problem, we propose to use the following score

$$s_{i,j} = p_{i,j} + p_{j,i}, \tag{20}$$

where $p_{i,j}$ is the pyramidal visibility score of Eq. (19) computed for image i only with the projections matched in image j. This score is more robust than the one in Eq. (18) as it takes into account both the overlap of the scene and the distribution of the projections in the views.

Acknowledgments

This work was partially supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468).

References

[1] 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.

[2] 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.

[3] P. F. Sturm and B. Triggs, “A factorization based algorithm for multi-image projective structure and motion,” in *European Conference on Computer Vision (ECCV)*, 1996, pp. 709–720.

[4] J. Oliensis and R. Hartley, “Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2217–2233, Dec 2007.

[5] S. Christy and R. Horaud, “Euclidean shape and motion from multiple perspective views by affine iterations,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 11, pp. 1098–1104, 1996.

[6] 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)*, ser. Lecture Notes in Computer Science, vol. 1406, 1998, pp. 296–310.

[7] A. Heyden, R. Berthilsson, and G. Sparr, “An iterative factorization method for projective structure and motion from image sequences,” *Image and Vision Computing*, vol. 17, no. 13, pp. 981–991, November 1999.

[8] Q. Chen and G. Medioni, “Efficient iterative solution to m-view projective reconstruction problem,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, vol. 2, 1999.

[9] 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*, vol. 1, 2001, pp. I–1018–I–1025.

[10] 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*, vol. 1, 2005, pp. 198–205.

[11] 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*, vol. 31, no. 5, pp. 841–854, 2009.

[12] Y. Dai, H. Li, and M. He, “Element-wise factorization for n-view projective reconstruction,” in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 6314, 2010, pp. 396–409.

[13] ——, “Projective multiview structure and motion from element-wise factorization,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 9, pp. 2238–2251, Sept 2013.

[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.

[15] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[16] 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)*, 2007, pp. 1–7.

[17] N. Gillis and F. Glineur, "Low-rank matrix approximation with weights or missing data is np-hard," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1149–1165, 2011.

[18] B. Triggs, "Factorization methods for projective structure and motion," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, Jun 1996, pp. 845–851.

[19] B. Nasihatkon, R. Hartley, and J. Trumpf, "A generalized projective reconstruction theorem and depth constraints for projective factorization," *International Journal of Computer Vision*, pp. 1–28, 2015.

[20] T. Wiberg, "Computation of principal components when data are missing," in *Proc. of Second Symp. Computational Statistics*, 1976, pp. 229–236.

[21] T. Okatani and K. Deguchi, "On the wiberg algorithm for matrix factorization in the presence of missing components," *International Journal of Computer Vision*, vol. 72, no. 3, pp. 329–337, 2007.

[22] D. Jacobs, "Linear fitting with missing data for structure-from-motion," *Computer Vision and Image Understanding*, vol. 82, no. 1, pp. 57 – 81, 2001.

[23] S. Olsen and A. Bartoli, "Using priors for improving generalization in non-rigid structure-from-motion," *Proc. British Machine Vision Conference*, 2007.

[24] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, and A. López, "An iterative multiresolution scheme for sfm," in *Image Analysis and Recognition*, A. Campilho and M. S. Kamel, Eds., 2006.

[25] V. Larsson, C. Olsson, E. Bylow, and F. Kahl, "Rank minimization with structured data patterns," in *European Conference on Computer Vision*. Springer, 2014, pp. 250–265.

[26] 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.

[27] A. Eriksson and A. Van Den Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l 1 norm," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 771–778.

[28] Q. Ke and T. Kanade, "Robust 1/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 739–746.

[29] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[30] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1087–1099, 2012.

[31] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[32] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 69–77.

[33] L. W. Mackey, M. I. Jordan, and A. Talwalkar, "Divide-and-conquer matrix factorization," in *Advances in Neural Information Processing Systems*, 2011, pp. 1134–1142.

[34] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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

[36] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *DAGM-Symposium*, 2003, pp. 236–243.

[37] P. Torr and A. Zisserman, "Mlesac," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, Apr. 2000.

[38] R. Hartley and F. Kahl, "Critical configurations for projective reconstruction from multiple views," *International Journal of Computer Vision*, vol. 71, no. 1, pp. 5–47, 2007.

[39] J. Nocedal and S. Wright, *Numerical Optimization*. Springer-Verlag New York, 2006.

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

[41] C. Olsson and O. Enqvist, "Stable structure from motion for unordered image collections," in *Image Analysis*, 2011, pp. 524–535.

[42] 3Dflow SRL, "3DF Zephyr reconstruction showcase," http://www.3dflow.net/, 2016.

[43] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *European Conference on Computer Vision*, 2010, pp. 29–42.

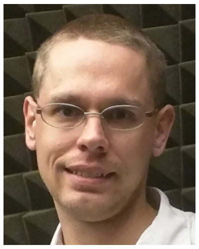
[44] P. Moulon and P. Monasse, "Unordered feature tracking made fast and easy," in *European Conference on Visual Media Production (CVMP)*, 2012.

[45] 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.

[46] 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*, vol. 2, 2005, pp. 316–322.

[47] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 6312, 2010, pp. 29–42.

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



Ludovic Magerand received the MSc and PhD degrees in computer science from Blaise Pascal University in 2008 and 2014 respectively. After being a postdoctoral fellow at the the Visual Geometry and Modelling (VGM) Lab of the Istituto Italiano di Tecnologia (IIT) in Genova for two years, he moved to teaching computer science in a French engineering school. He is currently a junior researcher at the Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) from the Czech Technical University (CTU) in Prague. He is working on geometric 3D vision: camera modelling, autocalibration, pose estimation and structure-from-motion.

Alessio Del Bue received the Laurea degree in Telecommunication engineering in 2002 from University of Genova and his Ph.D. degree in Computer Science from Queen Mary University of London in 2006. He was a researcher in the Institute for Systems and Robotics (ISR) at the Instituto Superior Tecnico (IST) in Lisbon, Portugal. Currently, he is leading the Visual Geometry and Modelling (VGM) Lab at the PAVIS department of the Istituto Italiano di Tecnologia (IIT) in Genova. His research focuses in the areas of 3D scene understanding and non-rigid structure from motion. He is a member of the IEEE.