Structure from Motion with Objects

Marco Crocco, Cosimo Rubino, Alessio Del Bue

Pattern Analysis and Computer Vision Department (PAVIS)
Visual Geometry and Modelling Lab (VGM)
Istituto Italiano di Tecnologia
Via Morego 30, 16163 Genova, Italy
alessio.delbue@iit.it

Abstract

This paper shows for the first time that is possible to reconstruct the position of rigid objects and to jointly recover affine camera calibration solely from a set of object detections in a video sequence. In practice, this work can be considered as the extension of Tomasi and Kanade factorization method using objects. Instead of using points to form a rank constrained measurement matrix, we can form a matrix with similar rank properties using 2D object detection proposals. In detail, we first fit an ellipse onto the image plane at each bounding box as given by the object detector. The collection of all the ellipses in the dual space is used to create a measurement matrix that gives a specific rank constraint. This matrix can be factorised and metrically upgraded in order to provide the affine camera matrices and the 3D position of the objects as an ellipsoid. Moreover, we recover the full 3D quadric thus giving additional information about object occupancy and 3D pose. Finally, we also show that 2D points measurements can be seamlessly included in the framework to reduce the number of objects required. This last aspect unifies the classical point-based Tomasi and Kanade approach with objects in a unique framework. Experiments with synthetic and real data show the feasibility of our approach for the affine camera case.

1. Introduction

Factorization methods for Structure from Motion (SfM) deliver highly efficient solutions for the simultaneous calibration and 3D reconstruction using image point trajectories/matches. The seminal paper of Tomasi and Kanade [19] was based on the intuition that if we form a matrix containing matched 2D image points, such resulting matrix is rank constrained. This property can be used to obtain an initial affine solution with Singular Value Decomposition (SVD) for the camera matrices and 3D points. Such solution can be then linearly upgraded to metric by imposing orthogonality constraints on the camera matrices, giving a closed-form solution to the SfM problem. Even if modern 3D reconstruction pipelines from images have reached impressive results through non-linear optimization [1, 8, 18], factorization methods still entails a theoretical appealing solution to the SfM problem. Such approaches have been further updated to more complex camera models [13, 21], to deal with the case of missing data [11, 17] and multiple moving objects [4, 6], or to model articulated [22, 20] and deformable [5]) objects, demonstrating that the research on this type of methods is still very active and promising. As a peculiar aspect, up to now, every factorization method for 3D reconstruction mostly deals with points, while very few exceptions exist in the literature using different geometrical entities such as lines [15, 14] and conic features [12, 16]. The limit of these methods is that they can reconstruct geometric primitives starting from projected outlines only, but they cannot deal with image object detections and their gross inaccuracies over size and position.

This work takes a different direction from previous SfM methods by showing that it is possible to solve simultaneously for affine camera calibration and 3D structure in a closed-form using multi-view relations given only by the location of a set of objects in an image sequence (see Fig 1 for a graphical representation). Instead of considering points as an input of our method, here we use as measurements the output of an object detector, i.e. a set of bounding boxes. We show that, by fitting the bounding boxes with 2D ellipsoids, it is possible to form a measurement matrix that contains the matrices of each 2D conics. If the collection of conics is expressed in the dual space, the resulting matrix $\mathtt{C}$ will show a specific rank constraint since the matrix of the dual conics can be decomposed in terms of two factors as $\mathtt{C} = \mathtt{G}\ \mathtt{V}$ where $\mathtt{G}$ contains the parameters of the affine camera and $\mathtt{V}$ the 3D quadrics whose reprojection gives the 2D

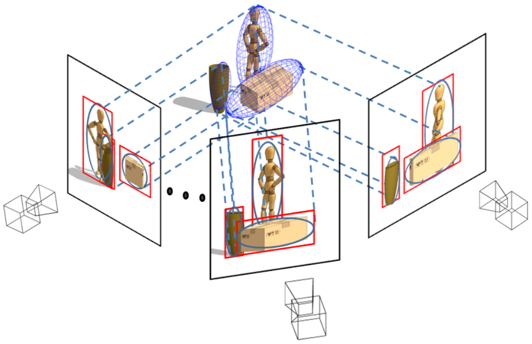

Figure 1. Given multiple views with a set of objects detected in every image, the proposed factorization approach can simultaneously recover the affine camera calibration and the 3D quadrics describing the location and pose of the objects in the scene.

conics stored in $\mathtt{C}$. Then, given this initial affine solution of the system, it is possible to find a metric upgrade that solves for both the camera matrices and the 3D quadrics. In practice this provides a self-calibration of the camera and 3D location of the objects with just a set of object detections. Moreover it is straightforward to show that image points can still be included in the formulation as degenerate conics in dual space thus obtaining a joint objects/points factorisation. Furthermore, since the framework recovers a 3D quadric related to an object, this information can be used to infer the coarse pose and size of the object shape in 3D. The rest of the paper is structured as follows. Section 2 defines the problem and the related mathematical formalisation. Section 3 presents the factorization problem in the dual space while Section 4 describes how to perform the metric upgrade. Experiments on real and synthetic data are discussed in Section 5 and then followed by concluding remarks in Section 6.

2. From bounding boxes to conics in dual space

Let us consider a set of image frames $f = 1 \ldots F$ representing a 3D scene under different viewpoints. A set of $i = 1 \ldots N$ rigid objects is placed in arbitrary positions and each object is detected in each of the F images. Each object i in each image frame f is identified by a 2D bounding box given by a generic object detector. In order to ease the mathematical formalization of the problem, we move from a bounding box representation of an object to an ellipsoid one. This is done by associating at each bounding box an ellipse fitting $\mathtt{D}_{fi}$ that inscribes the bounding box. The aim of our problem is to find the 3D ellipsoids $\mathtt{E}_i$ whose projections onto the image planes, associated to each frame $f = 1 \ldots F$, best fit the 2D ellipses $\mathtt{D}_{fi}$. This will solve for both the 3D localisation and occupancy of each object starting from image detections in the different views. In the following, we represent each ellipse using the homogeneous quadratic form of a conic equation:

$$\mathbf{u}^\top \mathtt{D}_{fi}\ \mathbf{u} = 0, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^3$ is the homogeneous vector of a generic 2D point belonging to the conic defined by the symmetric matrix $\mathtt{D}_{fi} \in \mathbb{R}^{3\times 3}$.

The conic has five degrees of freedom, given by the six elements of the lower triangular part of the symmetric matrix $\mathtt{D}_{fi}$ except one for the scale, since Eq. (1) is homogeneous in $\mathbf{u}$. Similarly to the ellipses, we represent the ellipsoids in the 3D space with the homogeneous quadratic form of a quadric equation:

$$\mathbf{x}^\top \mathtt{E}_i\, \mathbf{x} = 0, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^4$ represents an homogeneous 3D point belonging to the quadric defined by the symmetric matrix $\mathtt{E}_i \in \mathbb{R}^{4\times 4}$. The quadric has nine degrees of freedom, given by the ten elements of the symmetric matrix $\mathtt{E}_i$ up to one for the overall scale.

Since the relationship between $\mathtt{D}_{fi}$ and $\mathtt{E}_i$ is not straightforward in the primal space, i.e. the Euclidean space of 3D points (2D points in the images), it is convenient to reformulate it in dual space, i.e. the space of the planes (lines in the images) [7]. In particular, the conics in 2D can be represented by the envelope of all the lines tangent to the conic curve, while the quadrics in 3D can be represented by the envelope of all the planes tangent to the quadric surface.

Hence, the dual quadric is defined by the matrix $\mathtt{Q}_i = adj(\mathtt{E}_i)$, where adj is the adjoint operator, and the dual conic is defined by $\mathtt{C}_{fi} = adj(\mathtt{D}_{fi})$ [9].

Each quadric $\mathtt{Q}_i$, when projected onto the image plane, gives a conic denoted with $\mathtt{C}_{fi} \in \mathbb{R}^{3\times 3}$. The relationship between $\mathtt{Q}_i$ and $\mathtt{C}_{fi}$ is defined by the orthographic projection matrix $\mathtt{P}_f \in \mathbb{R}^{3\times 4}$ as:

$$\mathtt{P}_f = \left[\begin{array}{c|c} \mathtt{R}_f & \mathbf{t}_f \\ \mathbf{0}_3^\top & 1 \end{array}\right] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\mathtt{R}_f \in \mathbb{R}^{2\times 3}$ is an orthographic camera matrix such that $\mathtt{R}_f\mathtt{R}_f^\top = \mathtt{I}_{2\times 2}$, the vector $\mathbf{t}_f$ is the camera translation and $\mathbf{0}_m$ denotes a vector of m zeros.

The dual conic $\mathtt{C}_{fi}$ and the dual quadric $\mathtt{Q}_i$ are defined up to an overall scale factor that can be arbitrarily fixed by setting the elements (3,3) of $\mathtt{C}_{fi}$ and (4,4) of $\mathtt{Q}_i$ to -1. After such normalization, the relation between a dual quadric and its dual conic projections can be written as:

$$\mathtt{C}_{fi} = \mathtt{P}_f \mathtt{Q}_i \mathtt{P}_f^\top. \tag{5}$$

3. Dual conic matrix factorization

In order to recover $\mathtt{Q}_i$ in closed form from the set of dual conics $\{\mathtt{C}_{fi}\}_{f=1\ldots F}$, we have to re-arrange Eq. (5)

$$\mathtt{G}_f = \begin{bmatrix} {p_{11}}^2 & 2\,p_{12}p_{11} & 2\,p_{13}p_{11} & 2\,p_{14}p_{11} & {p_{12}}^2 & 2\,p_{13}p_{12} & 2\,p_{14}p_{12} & {p_{13}}^2 & 2\,p_{13}p_{14} & {p_{14}}^2 \\ p_{21}p_{11} & p_{21}p_{12}+p_{22}p_{11} & p_{23}p_{11}+p_{21}p_{13} & p_{24}p_{11}+p_{21}p_{14} & p_{22}p_{12} & p_{22}p_{13}+p_{23}p_{12} & p_{22}p_{14}+p_{24}p_{12} & p_{23}p_{13} & p_{23}p_{14}+p_{24}p_{13} & p_{24}p_{14} \\ 0 & 0 & 0 & p_{11} & 0 & 0 & p_{12} & 0 & p_{13} & p_{14} \\ {p_{21}}^2 & 2\,p_{22}p_{21} & 2\,p_{23}p_{21} & 2\,p_{24}p_{21} & {p_{22}}^2 & 2\,p_{23}p_{22} & 2\,p_{24}p_{22} & {p_{23}}^2 & 2\,p_{23}p_{24} & {p_{24}}^2 \\ 0 & 0 & 0 & p_{21} & 0 & 0 & p_{22} & 0 & p_{23} & p_{24} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

into a linear system. Let us define $\mathbf{v}_i = vech(\mathtt{Q}_i)$ and $\mathbf{c}_{fi} = vech(\mathtt{C}_{fi})$ as the vectorization of symmetric matrices $\mathtt{Q}_i$ and $\mathtt{C}_{fi}$ respectively[1].

Then, let us arrange the products of the elements of $\mathtt{P}_f$ and $\mathtt{P}_f^\top$ in a unique matrix $\mathtt{G}_f \in \mathbb{R}^{6\times 10}$ as follows [10]:

$$\mathtt{G}_f = \mathtt{Y}(\mathtt{P}_f \otimes \mathtt{P}_f)\mathtt{W} \quad (6)$$

where $\otimes$ is the Kronecker product and matrices $\mathtt{Y} \in \mathbb{R}^{6\times 9}$ and $\mathtt{W} \in \mathbb{R}^{16\times 10}$ are two matrices such that $vech(\mathtt{X}) = \mathtt{Y}\; vec(\mathtt{X})$ and $vec(\mathtt{X}) = \mathtt{W}\; vech(\mathtt{X})$ respectively, where $\mathtt{X}$ is a symmetric matrix[2]. Given $\mathtt{G}_f$, we can rewrite Eq. (5) as:

$$\mathbf{c}_{fi} = \mathtt{G}_f \mathbf{v}_i, \quad (7)$$

The set of equations for each frame and object can be rewritten in a global matrix system by stacking all the equations for each frame and view such that:

$$\mathtt{C} = \begin{bmatrix} \mathbf{c}_{11} & \cdots & \mathbf{c}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{c}_{F1} & \cdots & \mathbf{c}_{FN} \end{bmatrix} = \begin{bmatrix} \mathtt{G}_1 \\ \vdots \\ \mathtt{G}_F \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_N \end{bmatrix} \quad (8)$$

with

$$\mathtt{G}^\top = \begin{bmatrix} \mathtt{G}_1^\top & \cdots & \mathtt{G}_F^\top \end{bmatrix}^\top \qquad \mathtt{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_N \end{bmatrix}, \quad (9)$$

with the dual conic matrix $\mathtt{C}$ being clearly rank constrained since being the product of two rank constrained matrices (i.e. $rank(\mathtt{C}) \leq 10$) given the dimensionality of the matrix factors $\mathtt{G}_{6F\times 10}$ and $\mathtt{V}_{10\times N}$.

The explicit form of $\mathtt{G}_f$, function of the entries p_{mn} of $\mathtt{P}_f$, as in Eq. 3, although complex, shows that the matrix is strongly structured. Now, starting solely from the image measurements, i.e. the dual conics stored in $\mathtt{C}$, it is now possible to obtain an initial, affine, solution by performing SVD over $\mathtt{C}$ and truncating to the first 10 components thus obtaining:

$$\mathtt{C} = \tilde{\mathtt{G}}\;\; \tilde{\mathtt{V}}. \quad (10)$$

This factorization is not unique, since it is possible to define a 10×10 full-rank transformation matrix such that $\mathtt{C} = \mathtt{G}\;\; \mathtt{V} = \tilde{\mathtt{G}}\mathtt{Z}\;\; \mathtt{Z}^{-1}\tilde{\mathtt{V}}$. Finding the transformation matrix $\mathtt{Z}$ that enforces the correct matrix structure of $\mathtt{G}$ or $\mathtt{V}$ is the core problem of any factorization methods.

4. Upgrading to metric

Unlike the classical Tomasi and Kanade factorization method, where $\mathtt{Z}$ is a 3×3 symmetric matrix, the solution of the transformation matrix in our problem has a higher dimensionality. However in the literature of factorization methods there are several examples with increasing complexity: Photometric stereo [3] solves for a 4×4 and 9×9 matrix while the non-rigid structure from motion problem [5] has an increasing dimensionality given the complexity of the shape (i.e. $3K \times 3K$ where K are the modes of deformation of the shape). Regardless this, our problem entails interesting differences that departs from the classical solution in structure from motion. In particular, we will show that there is a feasible solution without computing the full $\mathtt{Z}_{10\times 10}$. First of all, the orthographic camera matrix constraints, i.e. $\mathtt{R}_f\mathtt{R}_f^\top = \mathtt{I}_{2\times 2}$, have to be enforced for the solution of the matrix $\mathtt{Z}$. However, the reshuffling of the components of $\mathtt{R}_f$ in the matrix $\mathtt{G}_f$ complicates further the problem. Yet, a key observation is that it is possible to re-arrange some of the entries of $\mathtt{G}_f$ in a new matrix that expresses a rank-3 constraint. This sub-problem can be solved and leads to the computation of the ortographic camera matrices, as well as translation parameters of the quadrics. Given this solution, by re-substituting into the original problem and with a careful normalization, it is possible to compute the quadrics shape and size linearly. Notice that this approach requires a minimum number of just three objects, linked to the rank 3 constraint, instead of ten objects required by the rank 10 factorization.

4.1. Solving for camera parameters and quadrics translation

A key step to reduce the complexity of the problem consists in enforcing a translation of each image frame according to the average of the ellipses coordinate centers. To do this, let us define $\mathbf{t}^c_{fi} \in \mathbb{R}^{2\times 1}$ as the center of ellipse i in frame f and $\mathbf{t}^q_i \in \mathbb{R}^{3\times 1}$ as the center of ellipsoid i in 3D space, and the related translation matrices $\mathtt{T}^c_{fi}$ and $\mathtt{T}^q_i$ as:

$$\mathtt{T}^c_{fi} = \left[\begin{array}{c|c} \mathtt{I}_{2\times 2} & \mathbf{t}^c_{fi} \\ \mathbf{0}_2^\top & 1 \end{array} \right], \qquad \mathtt{T}^q_i = \left[\begin{array}{c|c} \mathtt{I}_{3\times 3} & \mathbf{t}^q_i \\ \mathbf{0}_3^\top & 1 \end{array} \right]. \quad (12)$$

[1] The operator $vech$ serializes the elements of the lower triangular part of a symmetric matrix, such that, given a symmetric matrix $\mathtt{X} \in \mathbb{R}^{n\times n}$, the vector $\mathbf{x}$, defined as $\mathbf{x} = vech(\mathtt{X})$, is $\mathbf{x} \in \mathbb{R}^g$ with $g = \frac{n(n+1)}{2}$.

[2] The operator vec serializes all the elements of a generic matrix.

$$\bar{\mathtt{G}}_f = \begin{bmatrix} {p_{11}}^2 & 2\,p_{12}p_{11} & 2\,p_{13}p_{11} & {p_{12}}^2 & 2\,p_{13}p_{12} & {p_{13}}^2 & 0 & 0 & 0 & 0 \\ p_{21}p_{11} & p_{21}p_{12}+p_{22}p_{11} & p_{23}p_{11}+p_{21}p_{13} & p_{22}p_{12} & p_{22}p_{13}+p_{23}p_{12} & p_{23}p_{13} & 0 & 0 & 0 & 0 \\ {p_{21}}^2 & 2\,p_{22}p_{21} & 2\,p_{23}p_{21} & {p_{22}}^2 & 2\,p_{23}p_{22} & {p_{23}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{11} & p_{12} & p_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{21} & p_{22} & p_{23} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

It is easy to demonstrate that:

$$(\bar{\mathtt{T}}^c_f)^{-1}\mathtt{C}_{fi}(\bar{\mathtt{T}}^c_f)^{-\top} = \bar{\mathtt{P}}_f(\bar{\mathtt{T}}^q)^{-1}\mathtt{Q}_i(\bar{\mathtt{T}}^q)^{-\top}\bar{\mathtt{P}}_f^\top, \tag{13}$$

where

$$\bar{\mathtt{T}}^c_f = \frac{1}{N}\sum_{i=1}^{N}\mathtt{T}^c_{fi}, \qquad \bar{\mathtt{T}}^q = \frac{1}{N}\sum_{i=1}^{N}\mathtt{T}^q_i, \tag{14}$$

$$\bar{\mathtt{P}}_f = \left[\begin{array}{c|c} \mathtt{R}_f & \mathbf{0}_2 \\ \mathbf{0}_3^\top & 1 \end{array}\right] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 \\ p_{21} & p_{22} & p_{23} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{15}$$

Thus, centering every frame on the average of ellipses centers is equivalent to center the 3D space on the average of ellipsoids centers, removing at the same time the translation components $\mathbf{t}_f$ of each projection matrix related to the cameras. Therefore, the relationship between quadrics and conics in Eq. (5) can be recast in the following way:

$$\bar{\mathtt{C}}_{fi} = \bar{\mathtt{P}}_f\bar{\mathtt{Q}}_i\bar{\mathtt{P}}_f^\top, \tag{16}$$

given that

$$\bar{\mathtt{C}}_{fi} = (\bar{\mathtt{T}}^c_f)^{-1}\mathtt{C}_{fi}(\bar{\mathtt{T}}^c_f)^{-\top}, \qquad \bar{\mathtt{Q}}_i = (\bar{\mathtt{T}}^q)^{-1}\mathtt{Q}_i(\bar{\mathtt{T}}^q)^{-\top}. \tag{17}$$

An interesting fact arises here: If we calculate the matrices $\mathtt{G}_f$ starting from $\bar{\mathtt{P}}_f$, according to Eq. (6), we see that all the entries of $\mathtt{G}_f$ containing p_{14} and p_{24} are zeroed, since $p_{14} = p_{24} = 0$ in Eq. (15). Now let us permute the matrix $\mathtt{G}_f$ taking its rows and columns in the order given by the index sets $\{1,2,4,3,5,6\}$ for the rows and $\{1,2,3,5,6,8,4,7,9,10\}$ for the columns, so defining a new matrix $\bar{\mathtt{G}}_f$ as in Eq. (11). The matrix $\bar{\mathtt{G}}_f$ has a block diagonal structure, in which the 3×6 upper left block groups all the entries quadratic in the p_{mn} terms, the 2×3 middle block groups the entries linear in p_{mn} and the lower right block is a constant scalar. Given this structure, the bilinear factorization problem based on the rank 10 constraint can be decoupled into two sub-problems with rank 6 and 3 constraints respectively. The last block of size 1×1 does not provide any additional information and can be discarded [3].

In particular, let us isolate the middle block of $\bar{\mathtt{G}}_f$, named here $\mathtt{G}^l_f$:

$$\mathtt{G}^l_f = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix}, \tag{18}$$

where the index l recalls the linearity of $\mathtt{G}^l_f$ with respect to the terms p_{ij}. Next, let us pick up the corresponding entries of vectors $\bar{\mathbf{c}}_{fi}$ and $\bar{\mathbf{v}}_i$, defined as $\bar{\mathbf{c}}_{fi} = vech(\bar{\mathtt{C}}_{fi})$ and $\bar{\mathbf{v}}_i = vech(\bar{\mathtt{Q}}_i)$, so obtaining the new reduced and permuted vectors:

$$\mathbf{v}^l_i = \bar{\mathbf{v}}_{i\{4,7,9\}}, \qquad \mathbf{c}^l_{fi} = \bar{\mathbf{c}}_{fi\{3,5\}}. \tag{19}$$

Finally let us group together all the frames and objects, similarly as Eqs. (8) and (9), obtaining the following matrices:

$$\mathtt{G}^{l\,\top} = \begin{bmatrix} \mathtt{G}^{l\,\top}_1 & \cdots & \mathtt{G}^{l\,\top}_F \end{bmatrix}^\top, \qquad \mathtt{V}^l = \begin{bmatrix} \mathbf{v}^l_1 & \cdots & \mathbf{v}^l_N \end{bmatrix}, \tag{20}$$

$$\mathtt{C}^l = \begin{bmatrix} \mathbf{c}^l_{11} & \cdots & \mathbf{c}^l_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{c}^l_{F1} & \cdots & \mathbf{c}^l_{FN} \end{bmatrix}. \tag{21}$$

In this way we end up with the following bilinear relationship:

$$\mathtt{C}^l = \mathtt{G}^l\ \mathtt{V}^l, \tag{22}$$

where $\mathtt{C}^l \in \mathbb{R}^{2F\times N}$, $\mathtt{G}^l \in \mathbb{R}^{2F\times 3}$ and $\mathtt{V}^l \in \mathbb{R}^{3\times N}$. By performing an SVD on $\mathtt{C}^l$ and truncating to the third singular value we obtain:

$$\mathtt{G}^l\mathtt{V}^l = \tilde{\mathtt{G}}^l\mathtt{Z}^l\quad \mathtt{Z}^{\mathtt{l}-1}\tilde{\mathtt{V}}^l. \tag{23}$$

The 3×3 mixing matrix $\mathtt{Z}^l$ can be found by exploiting the orthogonality and norm constraints of an orthographic camera matrix. In this way we are able to find the whole camera parameters and the elements 4, 7 and 9 of the vectorized quadrics, given the measured elements 3 and 5 of the vectorized conics, by simply selecting the proper elements of the recovered matrices $\mathtt{G}^l$ and $\mathtt{V}^l$.

Geometrical interpretation. In order to gain more insights into the geometrical meaning of this solution, the relation between conics and quadrics will be made explicit. Every dual conic $\mathtt{C}_{fi}$ can be expressed as a conic with coordinate center $(0,0)$, denoted with $\breve{\mathtt{C}}_{fi}$, pre- and post-multiplied by a translation matrix $\mathtt{T}^c_{fi}$ defined according to Eq. (12). A similar property holds for quadrics, thus giving:

$$\mathtt{C}_{fi} = \mathtt{T}^c_{fi}\breve{\mathtt{C}}_{fi}\mathtt{T}^{c\top}_{fi}, \qquad \mathtt{Q}_i = \mathtt{T}^q_i\breve{\mathtt{Q}}_i\mathtt{T}^{q\top}_i, \tag{24}$$

[3] Given that the last elements of all quadrics and conics are fixed to -1, it is easy to see that the last row of $\bar{\mathtt{G}}_f$ provides a redundant constraint -1 = -1 for every frame and object.

where:

$$\breve{\mathtt{C}}_{fi} = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{22} & 0 \\ 0 & 0 & \text{-1} \end{bmatrix}, \qquad \breve{\mathtt{Q}}_i = \begin{bmatrix} q_{11} & q_{12} & q_{13} & 0 \\ q_{12} & q_{22} & q_{23} & 0 \\ q_{13} & q_{23} & q_{33} & 0 \\ 0 & 0 & 0 & \text{-1} \end{bmatrix}. \tag{25}$$

Given Eqs. (24) and (25), the vectorized dual conics and dual quadrics assume the following form:

$$\mathbf{c}_{fi} = \begin{bmatrix} c_{11} - {t^c_1}^2 \\ c_{12} - t^c_1 t^c_2 \\ -t^c_1 \\ c_{22} - {t^c_2}^2 \\ -t^c_2 \\ -1 \end{bmatrix} \qquad \mathbf{v}_i = \begin{bmatrix} q_{11} - {t^q_1}^2 \\ q_{12} - t^q_1 t^q_2 \\ q_{13} - t^q_1 t^q_3 \\ -t^q_1 \\ q_{22} - {t^q_2}^2 \\ q_{23} - t^q_2 t^q_3 \\ -t^q_2 \\ q_{33} - {t^q_3}^2 \\ -t^q_3 \\ -1 \end{bmatrix}. \tag{26}$$

Notice that the translation parameters t^c_1, t^c_2 and t^q_1, t^q_2, t^q_3 of conic and quadric appear linearly in entries 3 and 5 of $\mathbf{c}_{fi}$ and 4, 7 and 9 of $\mathbf{v}_i$. Since these entries are the ones picked up to form the matrices $\mathtt{C}^l$ and $\mathtt{V}^l$ we can conclude that ellipsoid centers can be recovered from ellipses centers by decoupling them from the ellipsoid shape.

4.2. Solving for the ellipsoid shape

As can be seen from Eqs. (26), the terms describing shape and size are contained in the elements $\{1, 2, 3, 5, 6, 8\}$ of the vectorized quadric $\mathbf{v}_i$ and the elements $\{1, 2, 4\}$ of the vectorized conic $\mathbf{c}_{fi}$. Therefore, it is possible to express the vectorized conics elements $\{1, 2, 4\}$ as a product of the 3×6 upper left block of $\bar{\mathtt{G}}_f$ times the vectorized quadrics elements $\{1, 2, 3, 5, 6, 8\}$ and exploit the rank-6 constraint to find such quadric elements. However three drawbacks make this option not so appealing.

First, the terms of the vectorized quadrics are mixed with the quadratic component related to the translation, as can be seen in Eq. (26). In the case of small ellipses far from the image center, a likely case in many scenarios, the terms related to the ellipse size and shape become negligible with respect to the translation terms, and consequently even small errors on $\mathbf{c}_{fi}$ affect negatively the reconstruction of the ellipsoid. This is because the information on the ellipsoid shape, embedded in the elements c_{11}, c_{12} and c_{22} do not prevail over the translation errors on t^c_1 and t^c_2. Second, by exploiting the rank-6 constraint, one can reconstruct the six ellipsoid terms up to a 6×6 invertible matrix. However doing a metric upgrade is not trivial involving quadratic equality constraints drawn from the structure of $\bar{\mathtt{G}}_f$, thus leading to a nonlinear optimization procedure with the risk to obtain a local solution. Finally, at least six objects have to be visible in every frame in order to be able to use a rank-6 constraint.

To fix these drawbacks, we propose a different procedure. First of all, it is possible to remove the quadratic translation terms by considering the centered ellipses and ellipsoids. In fact, if Eq. (16) holds, then also:

$$\breve{\mathtt{C}}_{fi} = \bar{\mathtt{P}}_f \breve{\mathtt{Q}}_i \bar{\mathtt{P}}_f^\top \tag{27}$$

holds, i.e. in the affine case centering every single ellipse to the image center is equivalent to center every ellipsoid in the 3D coordinates origin. The vectorized versions of centered conics and quadrics, defined as $\check{\mathbf{v}}_i = vech(\breve{\mathtt{Q}}_i)$ and $\check{\mathbf{c}}_{fi} = vech(\breve{\mathtt{C}}_{fi})$ do not contain translation terms, while the shape terms c_{mn} and q_{mn} have been left unchanged.

Therefore we can select the rows and columns accounting for shape in $\bar{\mathtt{G}}_f$, $\check{\mathbf{v}}_i$ and $\check{\mathbf{c}}_{fi}$ as follows:

$$\mathtt{G}^s_f = \bar{\mathtt{G}}_{f\{1,2,3\}\times\{1,2,3,4,5,6\}}, \tag{28}$$

$$\mathbf{v}^s_i = \check{\mathbf{v}}_{i\{1,2,3,5,6,8\}}, \qquad \mathbf{c}^s_{fi} = \check{\mathbf{c}}_{fi\{1,2,4\}} \tag{29}$$

and build the matrices:

$${\mathtt{G}^s}^\top = \left[{\mathtt{G}^s_1}^\top \;\; \cdots \;\; {\mathtt{G}^s_F}^\top \right]^\top, \quad \mathtt{V}^s = \left[\mathbf{v}^s_1 \;\; \cdots \;\; \mathbf{v}^s_N \right], \tag{30}$$

$$\mathtt{C}^s = \begin{bmatrix} \mathbf{c}^s_{11} & \cdots & \mathbf{c}^s_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{c}^s_{F1} & \cdots & \mathbf{c}^s_{FN} \end{bmatrix}, \tag{31}$$

where $\mathtt{C}^s \in \mathbb{R}^{3F\times N}$, $\mathtt{G}^s \in \mathbb{R}^{3F\times 6}$ and $\mathtt{V}^s \in \mathbb{R}^{6\times N}$. The three matrices are linked by the bilinear relation:

$$\mathtt{C}^s = \mathtt{G}^s \mathtt{V}^s. \tag{32}$$

At this point, instead of performing an SVD, we can exploit the camera parameters p_{mn} found by the rank 3 solution in Eq. (23), to recover the matrix $\mathtt{G}^s$ by simple variable assignments, exploiting the structure in Eq. (11). Next, we can estimate the shape and size parameters independently for each quadric, multiplying the pseudo inverse of $\mathtt{G}^s$ by each column of $\mathtt{C}^s$:

$$\mathtt{V}^s = \mathtt{G}^{s+} \mathtt{C}^s \tag{33}$$

where $\mathtt{G}^{s+}$ is the pseudo inverse of $\mathtt{G}^s$. Finally, we recombine the shape and size parameters contained in $\mathtt{V}^s$ with the translation parameters in matrix $\mathtt{V}^l$ in Eq. (23), recovering the correct ellipsoids.

4.3. Solution with both points and objects

Interestingly, it is possible to include both point matches and objects in the same factorization framework. This might be convenient when the number of objects is not enough or to make more accurate the estimation with additional information from reliable 2D tracks.

To this end, P points can be included in this framework by expressing them as P additional degenerate quadrics or conics. In particular, we associate to each point an arbitrary quadric (and a set of conics in images) whose center is equal to the point coordinates, and then we evaluate the limit for the size of quadric and conic going to zero. In detail, the P additional quadrics and conics in function of a size parameter h are defined as:

$$\mathtt{C}_{f,N+p}(h) = \mathtt{T}^{c}_{f,N+p}\mathtt{H}^{c}\breve{\mathtt{C}}_{f,N+p}\mathtt{H}^{c\top}\mathtt{T}^{c\top}_{f,N+p} \tag{34}$$

$$\mathtt{Q}_{N+p}(h) = \mathtt{T}^{q}_{N+p}\mathtt{H}^{q}\breve{\mathtt{Q}}_{N+p}\mathtt{H}^{q\top}\mathtt{T}^{q\top}_{N+p} \tag{35}$$

for $p = 1, \dots P$, with:

$$\mathtt{H}^{c} = \begin{bmatrix} h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathtt{H}^{q} = \begin{bmatrix} h & 0 & 0 & 0 \\ 0 & h & 0 & 0 \\ 0 & 0 & h & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{36}$$

Notice that the point location information is given by the translation matrices $\mathtt{T}^{c}_{f,N+p}$ and $\mathtt{T}^{q}_{N+p}$. The degenerate conics and quadrics that correspond to a point can be written as:

$$\mathtt{C}_{f,N+p} = \lim_{h\to 0} \mathtt{C}_{f,N+p}(h) \qquad \mathtt{Q}_{N+p} = \lim_{h\to 0} \mathtt{Q}_{N+p}(h) \tag{37}$$

Notice that in the corresponding vectorized quadrics and conics the shape terms c_{mn} and q_{mn} are multiplied by h and consequently vanish, leaving just the translation terms. At this point the same approach described in Section 4.1 can be followed, simply adding to the number of objects N the number of points $N + P$. Differently, once the camera parameters, together with 3D points and ellipsoids centers have been found, the matrix related to ellipsoids shape will have the same dimension of $\mathtt{C}^{s}$ in Eq. (32), i.e. $3F \times N$ since the additional P columns related to the points contain just zeros and can be removed.

5. Experiments

The proposed method has been tested on a synthetic scenario and a real dataset. In every experiment, the accuracy of the estimated 3D object position and pose was measured by the volume overlap O_{3D} given by the intersection between ground truth (GT) and estimated (ES) ellipsoids respectively:

$$O_{\text{3D}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathcal{Q}_i \cap \tilde{\mathcal{Q}}_i}{\mathcal{Q}_i \cup \tilde{\mathcal{Q}}_i}, \tag{38}$$

where $\mathcal{Q}_i$ and $\tilde{\mathcal{Q}}_i$ denote the volume of GT and ES ellipsoids respectively. This metric measures the success of the algorithm in recovering the 3D position and occupancy of an object. Moreover, we also evaluated the orientation error by using the measure θ_{err}, which is the angle in radians between the main axes of GT and ES ellipsoids. We tested the proposed algorithm performance in two conditions: using ellipses from bounding boxes only (ELL) and using ellipses plus a set of additional points ($ELL + P$).

5.1. Synthetic setup

We generated a synthetic 3D setup with a variable number of ellipsoids, randomly placed inside a cube of side 20 units. The length of the largest ellipsoid axis L ranges from 3 to 12 units, according to a uniform PDF. The lengths of the other two axes are equal to γL with $\gamma \in [0.3, 1]$. Finally, axes orientation was fixed randomly. Optionally, we added a variable number of 3D points randomly generating their positions. A set of 20 camera views were generated and the camera trajectory was computed so that azimuth and elevation angles span the range $[0°, 60°]$ and $[0°, 70°]$ respectively. Given the orthographic camera matrix $\mathtt{P}_f$ of each camera frame, GT ellipses and GT 2D points were calculated from the exact projections of the ellipsoids and 3D points.

Synthetic tests were aimed at validating the robustness of the proposed method against common inaccuracies affecting object detectors, such as coarse estimation of the object center, tightness of the bounding box with respect to the object size and variations over the object pose. Thus, each ellipse was corrupted by three errors, namely translation error (TE), rotation error (RE) and size error (SE). To impose such errors, the ellipses centers coordinates t^c_1, t^c_2, the axes length l_1, l_2 and the orientation α of the first axis were perturbed as follow[4]:

$$\hat{t}^c_j = t^c_j + \bar{l}\nu^{\mathbf{t}}_j, \qquad \hat{\alpha} = \alpha + \nu^{\alpha}, \qquad \hat{l}_j = l_j\left(1 + \nu^{l}\right), \tag{39}$$

where $\nu^{\mathbf{t}}_j$, ν^{α} and ν^{l} are random variables with uniform PDF and mean value equal to zero, and $\bar{l} = (l_1 + l_2)/2$. Translation errors were also imposed to 3D points, with a magnitude calculated according to Eq. (39), assuming a random ellipsoid associated to each point. In order to highlight the specific impact of each error, they were applied separately. Error magnitudes were set tuning the boundary values of the uniform PDFs of $\nu^{\mathbf{t}}_j$, ν^{α} and ν^{l}. In detail, for each kind of error, we considered 10 different values of $\nu^{\mathbf{t}}_j$, ν^{a} and ν^{l}, with uniform spacing, and we applied the resulting error realizations to the ellipses reprojections related to all the ellipsoid. We run 100 trials for each setup, described by the number of objects and error on ellipses.

In Fig. 2, O_{3D} and θ_{err} are displayed versus RE and SE. Reconstruction is perfect for zero errors, with $O_{3D} = 1$,

[4] We omit for simplicity the object and frame indexes.

$\theta_{err} = 0$ and smoothly worsening as the error increases, reaching a minimum $O_{3D} = 0.5$ for RE $= 45°$ or SE = $= 0.5$. The pose error θ_{err} reaches a maximum value of about $50°$ for RE $= 45°$, and $40°$ for SE $= 0.5$. Overall, results appear to be quite robust to SE and RE. This fact is particularly important since such errors are likely to happen very frequently whenever ellipses are fitted to BBs. Even if the detector is accurate, the bounding box quantises the object alignment at steps of 90°, yielding a maximum RE of 45°. This tends to overestimate the object area, thus affecting SE, whenever the object is not aligned to the bounding box axes.

Notice that the performance does not vary with the number of ellipsoids present in the scene. Though this may seem counterintuitive, it follows from the fact that RE and SE do not affect the estimation of camera parameters, the latter being based on translation terms only, as can be seen from Eq. (23). Once camera parameters are recovered, the ellipsoid shape and size is estimated separately for each object (check Eq. (33)), so making the overall performance simply the average of the performance for each single object. The above reasoning is not valid for TE, as can be seen in Fig. 3. Both O_{3D} and θ_err are dependent on the number of objects in the scene, showing a notable increase in performance passing from 4 to 7 objects and a further slight increase with 10 objects.

Adding points to the setup improves in general the results and allows to overcome the minimal requirement of 3 objects. In Fig. 4 we report the results versus TE for 2, 3 and 4 objects with 2 and 5 additional points. In general, what matters for the performance versus TE is the sum of numbers of objects and points. In particular the performance grows strongly starting from 4 points/objects and tends to saturate above about 10 points/objects. Differently the performance versus RE and SE are not reported, since they are independent from the number of points and very similar to the case without objects (Fig. 2).

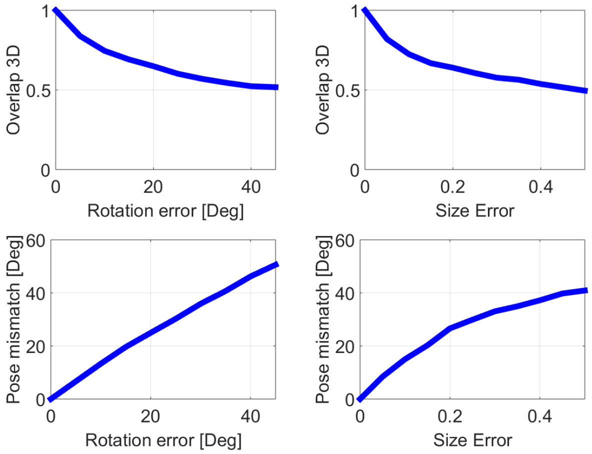

Figure 2. Results for the synthetic tests without additional points versus RE and SE errors. First row: 3D overlap; second row: pose mismatch; first column: RE; second column: SE.

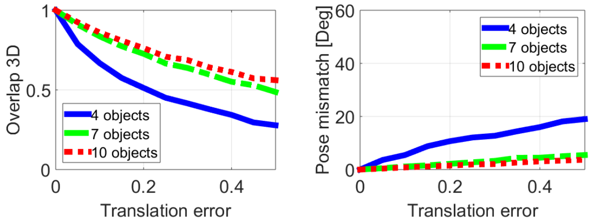

Figure 3. Results for the synthetic tests with 4, 7 and 10 objects without additional points, versus TE. Left: 3D overlap; right: pose mismatch.

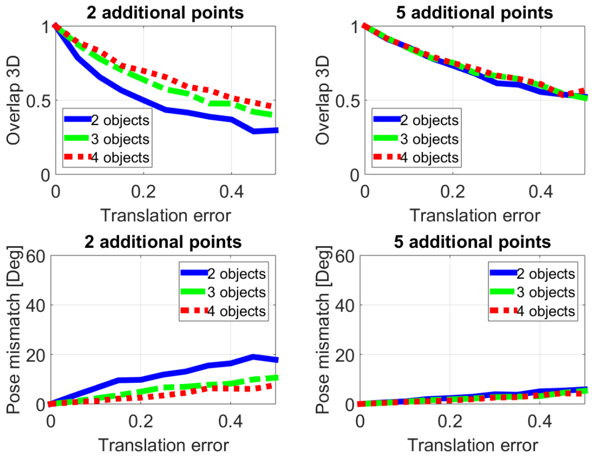

Figure 4. Results for the synthetic tests versus RE for 2, 3 and 4 objects with 2 or 5 additional points. First row: 3D overlap; second row: pose mismatch; first column 2 points; second column: 5 points.

5.2. Real setup: KINECT dataset

We tested the proposed algorithm on the KINECT dataset [2]. The dataset is composed of five sequences, each one showing a different office desk, with about $10 - 15$ objects, from a variable number of frames (but always less than a hundred). Bounding boxes associated to each objects are also provided. The dataset is very challenging as the angle spanned by the camera views is quite narrow. Moreover bounding boxes are quite unprecise in terms of position and aspect ratio. We selected a subset of about $8 - 25$ frames for each sequence, associating an ellipse to each BB at each frame. We also extracted a set of points tracks and run the proposed algorithm on ELL and $ELL + P$ setups. In Fig. 5 we show the results for Seq. 2, 3 and 4 (ELL). Reprojected ellipses match very well the position, shape and pose of all the objects in all the frames, as exemplified by the three image frames in Fig. 5. More importantly, the object's relative displacement along the z direction, visible in the upper views of 3D reconstructions, is correctly estimated as well as the objects size. Moreover, for the majority of objects, the aspect ratio and pose is also qualita-

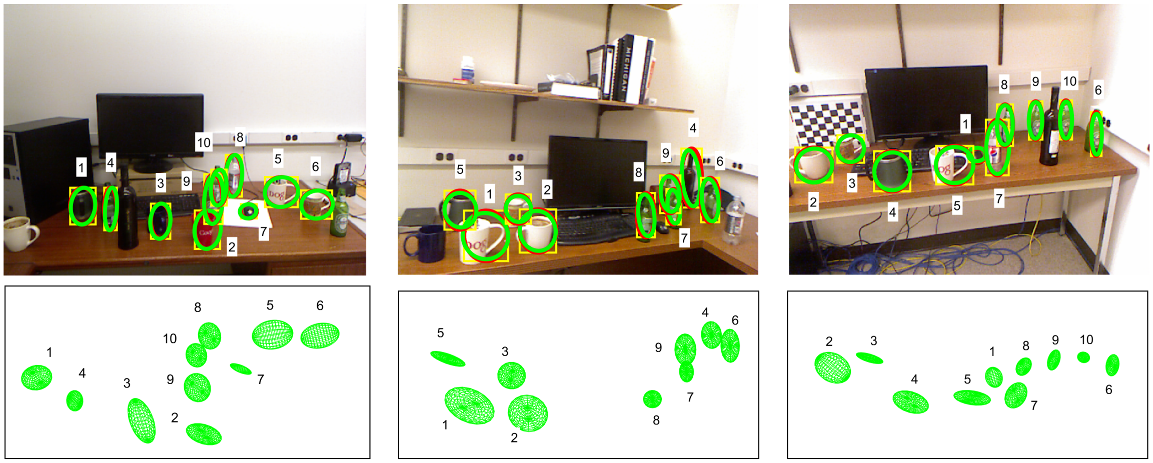

Figure 5. Results for the KINECT dataset for ELL setup on Sequences 2 (first column), 3 (second column) and 4 (third column.) First row: a frame from the sequence with BBs (yellow), ellipses from BBs (red) and reprojected ellipses (green) . Second row: upper views of the ES ellipsoids.

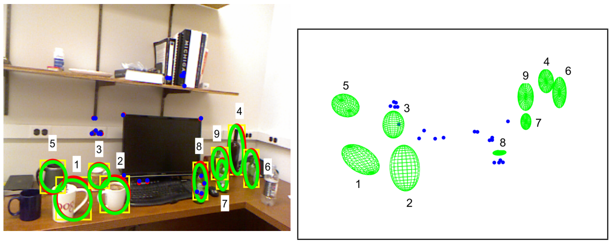

Figure 6. Results for the KINECT dataset for $ELL + P$ setup on Sequence 3. Left: a frame from the sequence with BBs (yellow), ellipses from BBs (red), points (red), reprojected ellipses (green) and points (blue). Right: upper views of the ES ellipsoids and 3D points.

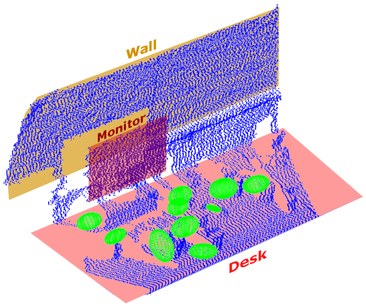

Figure 7. 3D estimated ellipsoids of Sequence 2 aligned to the KINECT point cloud.

tively correct. Some exceptions like the mugs $2, 3$ in Seq. 2 and $3, 4, 5$ in Sequence 4 are due to their asymmetric shape for which an ellipsoid represent an intrinsically coarse approximation. Sequence 3 was also tested in the $ELL + P$ setup, adding about 25 point tracks. The result reported in Fig. 6 is structurally very similar to the corresponding ELL case for the objects, though ellipsoids are more stretched along the z direction. Also the 3D structure of the points is correctly recovered and it is coherent with the objects displacement. For example, points belonging to object 8 are very close to the corresponding ellipsoid, while points from the PC monitor and from the books on the shelf are located at the correct depth with respect to all the objects. These results witness the capability of the method to solve a generalized SfM problem, embedding in a single framework both quadrics and points. Finally, to better understand the goodness of the objects 3D estimation, Fig. 7 shows the ellipsoid position being coherent with the point cloud: The different objects are lying onto the table and with a correct depth. Some inaccuracies are present for lateral objects due to coarse depth estimates and the possible discrepancy between the perspective model of the Kinect camera and the orthographic one assumed in our method.

6. Conclusions

A generalized SfM method has been proposed that is able to recover, in closed-form, camera poses and objects positions in 3D space, taking as input just a set of bounding boxes from an object detector in a collection of image frames. The devised method, exploiting the relationship between quadrics and conics in dual space, is able to recover also a coarse estimation of objects size, shape and orientation in space. Finally, 2D points can be easily included in the framework as degenerate conics, so overcoming the constraints on the minimal number of objects required. The method has been tested on both synthetic and real data, demonstrating its robustness against possible object detector inaccuracies, and proving to be effective in real challenging conditions. Further research on this topic will include solution refinement methods based on non-linear cost functions able to solve in one step for all the unknowns. Moreover, given the new ideas introduced in this work, we will deal with the more general and complex case of perspective SfM using dual matrix factorization with conics.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *ECCV 2010*, pages 29–42. Springer, 2010. 1

[2] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR 2011*, pages 2025–2032. IEEE, 2011. 7

[3] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *IJCV 2007*, 72(3):239–257, 2007. 3

[4] T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 179–186. IEEE, 1991. 1

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR 2000*, volume 2, pages 690–696. IEEE, 2000. 1, 3

[6] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *ICCV 1995*, pages 1071–1076. IEEE, 1995. 1

[7] G. Cross and A. Zisserman. Quadric reconstruction from dual-space geometry. In *ICCV 1998*, pages 25–31. IEEE, 1998. 2

[8] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. In *ECCV 2010*, pages 368–381. Springer, 2010. 1

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 2

[10] H. V. Henderson and S. Searle. Vec and vech operators for matrices, with some uses in jacobians and multivariate statistics. *Canadian Journal of Statistics*, 7(1):65–81, 1979. 3

[11] D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *CVPR 1997*, pages 206–212. IEEE, 1997. 1

[12] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *IJCV 1999*, 33(3):163–180, 1999. 1

[13] K. Kanatani and Y. Sugaya. Factorization without factorization: complete recipe. *Mem. Fac. Eng. Okayama Univ*, 38(1&2):61–72, 2004. 1

[14] D. Matinec and T. Pajdla. Line reconstruction from many perspective images by factorization. In *CVPR 2003*, volume 1, pages I–497. IEEE, 2003. 1

[15] L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *CVPR 1996*, pages 803–808. IEEE, 1996. 1

[16] L. Reyes and E. Bayro-Corrochano. The projective reconstruction of points, lines, quadrics, plane conics and degenerate quadrics using uncalibrated cameras. *Image and Vision Computing*, 23(8):693–706, 2005. 1

[17] H.-Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *PAMI 1995*, 17(9):854–867, 1995. 1

[18] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello. Hierarchical structure-and-motion recovery from uncalibrated images. *CVIU 2015*, 140:127 – 143, 2015. 1

[19] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV 1992*, 9(2):137–154, 1992. 1

[20] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *CVPR 2005*, volume 2, pages 1110–1115. IEEE, 2005. 1

[21] B. Triggs. Factorization methods for projective structure and motion. In *CVPR 1996*, pages 845–851. IEEE, 1996. 1

[22] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV 2006*, pages 94–106. Springer, 2006. 1