International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Estimation of 3D Category-Specific Object Structure: Symmetry, Manhattan and/or Multiple Images

Yuan Gao · Alan L. Yuille

Received: date / Accepted: date

Abstract Many man-made objects have intrinsic symmetries and often Manhattan structure. By assuming an orthographic or weak perspective projection model, this paper addresses the estimation of 3D structures and camera projection using symmetry and/or Manhattan structure cues, for the two cases when the input is a single image or multiple images from the same category, *e.g.* multiple different cars from various viewpoints. More specifically, analysis on the single image case shows that Manhattan alone is sufficient to recover the camera projection and then the 3D structure can be reconstructed uniquely by exploiting symmetry. But Manhattan structure can be hard to observe from a single image due to occlusion. Hence, we extend to the multiple image case which can also exploit symmetry but does not require Manhattan structure. We propose novel structure from motion methods for both rigid and non-rigid object deformations, which exploit symmetry and use multiple images from the same object category as input. We perform experiments on the Pascal3D+ dataset with either human labeled 2D keypoints or with 2D keypoints localized from a convolutional neural network. The results show that our methods which exploit symmetry significantly outperform the baseline methods.

Keywords Symmetry · Manhattan · Single Image · Symmetric Rigid Structure from Motion · Symmetric Non-Rigid Structure from Motion

Yuan Gao
Tencent AI Lab
E-mail: ethan.y.gao@gmail.com

Alan L. Yuille
Johns Hopkins University and UCLA
E-mail: alan.yuille@jhu.edu

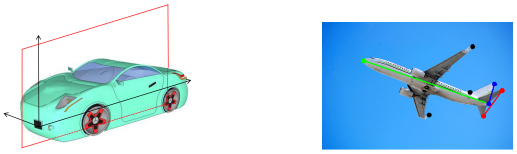

Fig. 1 Left Panel: Illustration of symmetry and Manhattan structure. The car has a bilateral symmetry with respect to the plane in red. There are three Manhattan axes. The first is normal to the symmetry plane of the car (*e.g.* from *left wheel* to *right wheel*). The second is from the front to the back of the car (*e.g. back left wheel* to *front right wheel*) while the third is in the vertical direction. Right Panel: Illustration of the 3 Manhattan directions on a real aeroplane image, shown by Red, Green, Blue lines. These 3 Manhattan directions can be obtained directly from the labeled keypoints.

1 Introduction

Many objects, especially those made by humans, have intrinsic symmetries [25, 39] and Manhattan structure (meaning that 3 perpendicular axes are inferable from the object [10, 11, 14]). These include cars and aeroplanes, see Fig 1. The purpose of this paper is to investigate the benefits of using symmetry and/or Manhattan constraints to estimate the 3D structures of objects from one or more images. As a key task in computer vision, numerous studies have been conducted on estimating the 3D shapes of objects from multiple images [1, 4, 12, 13, 15, 16, 18, 22, 23, 43, 44, 48]. There is also a long history of research on the use of symmetry [17, 25, 27, 29, 35, 41, 45] and a growing body of work on Manhattan world [10, 11, 14]. There is, however, little work that combines these cues.

This paper was inspired by recent work [26, 46], which estimates the 3D structure of an object class using structure from motion (SfM), taking multiple intra-class instances as input, *e.g.* different cars from vari-

ous viewpoints, but which did not exploit symmetry or Manhattan constraints. Following [26, 46], we use the 2D positions of keypoints as input to estimate the 3D structure and the camera projection, leaving the detection of the 2D keypoints to methods such as [9]. In this paper, different combinations of the three cues, *i.e.* symmetry, Manhattan and multiple images, are investigated. We propose four new algorithms for estimating 3D structure, assuming orthographic or weak perspective (*i.e.* orthographic projection plus scale) projection. These include an algorithm for single image reconstruction using both symmetry and Manhattan constraints, an algorithm for multiple image reconstruction using symmetry for objects with rigid deformations, and two for multiple image reconstruction exploiting symmetry for objects with non-rigid deformations. We experimented with using Manhattan, in addition to symmetry, for multiple images but found it gave negligible improvement. Note that rigidity or non-rigidity in this paper means the deformation between the objects from the same category, *e.g.* between the sedan and SUV cars, is assumed to be rigid or non-rigid, but the objects themselves are rigid and symmetric.

We start by explaining our core strategy for exploiting symmetry assuming all keypoints are observed. We exploit symmetry by a coordinate rotation which enables us to decouple the estimation of the 3D structure so that different components of the 3D structure can be estimated separately. This enables us to adopt the standard factorization methods for rigid and non-rigid motion to take advantage of symmetry. Specifically for the rigid case, it enables us to reduce the problem to applying Singular Value Decomposition (SVD) twice, to estimate different 3D structure components, and then to identify and to combine these estimates while dealing with the ambiguities resulting from SVD. The same coordinate rotation can be applied to estimating 3D structure from single images or for multiple images for both rigid and non-rigid deformations. This strategy explains the close relationships between earlier versions of this work which appeared in two conference papers [15, 16].

Next, we proceed to the single image reconstruction case. This exploits both the symmetry and Manhattan constraints, see Fig. 1. We show that Manhattan alone is sufficient to estimate the camera projection (*i.e.* the viewpoint of the object). Then symmetry is used to estimate the 3D structure by exploiting the change of coordinates mentioned in the previous paragraph. We illustrate reconstruction from single images using aeroplanes from Pascal3D+, see the experimental section. But we found it hard to estimate the Manhattan axes from a single image, due to occlusion of keypoints, so we only report results for a limited number of cases.

As it is impractical to assume all the keypoints are visible (as assumed in the single image reconstruction), we move on to exploit multiple-images to deal with occlusions and obtain better estimates of the 3D structure. We formulate the problem in terms of energy minimization with symmetry constraints included. The energy includes missing/latent variables to deal with unobserved keypoints due to occlusion. These complications mean that we cannot directly apply factorization methods(e.g., SVD as in previous work [28, 42]) to directly minimize the energy function. Instead, we use coordinate descent but, in order to give a good initialization, we define a *surrogate energy function* which exploits symmetry, by grouping the keypoints into symmetric keypoint pairs, and assumes that the missing data are known (*e.g.* initialized by another process, or estimated iteratively like EM). This procedure applies to all of our Structure from Motion (SfM) methods, although they differ in how to estimate good initializations for 3D structure and viewpoint.

After that, we discuss the details of our symmetric SfM method assuming rigid deformation. The input consists of different intra-class object instances, seen from a variety of viewpoints, and we assume rigid deformation between different intra-class instances. Combining these estimates requires analyzing the ambiguities inherent in SVD and specifying algorithms to resolve them. We call this method *Sym-RSfM*. It is tested and compared to baseline methods, which do not exploit symmetry, on benchmarked datasets.

We extend our approach to the non-rigid case and propose two symmetric non-rigid SfM methods making the standard assumption that the 3D structures can be represented as linear combinations of basis function (where the coefficients vary for different objects). The first approach is a direct factorization method, which extends the prior-free approach [12, 13] and exploits symmetry by a coordinate transformation to decompose the 3D structure into independent components while addressing the complex ambiguities which arise in factorization approaches to the non-rigid structure from motion (and which are modified due to our use of symmetry). Our second approach is an extension of [44] which uses a Gaussian prior on the coefficients of the deformation bases and an EM algorithms (note that recent work [2] showed that the prior was not necessary, thereby motivating the prior-free methods). We refer to these two methods as *Sym-PriorFree* and *Sym-EM-PPCA* respectively. In the experimental section, we compare their performance to the corresponding base-

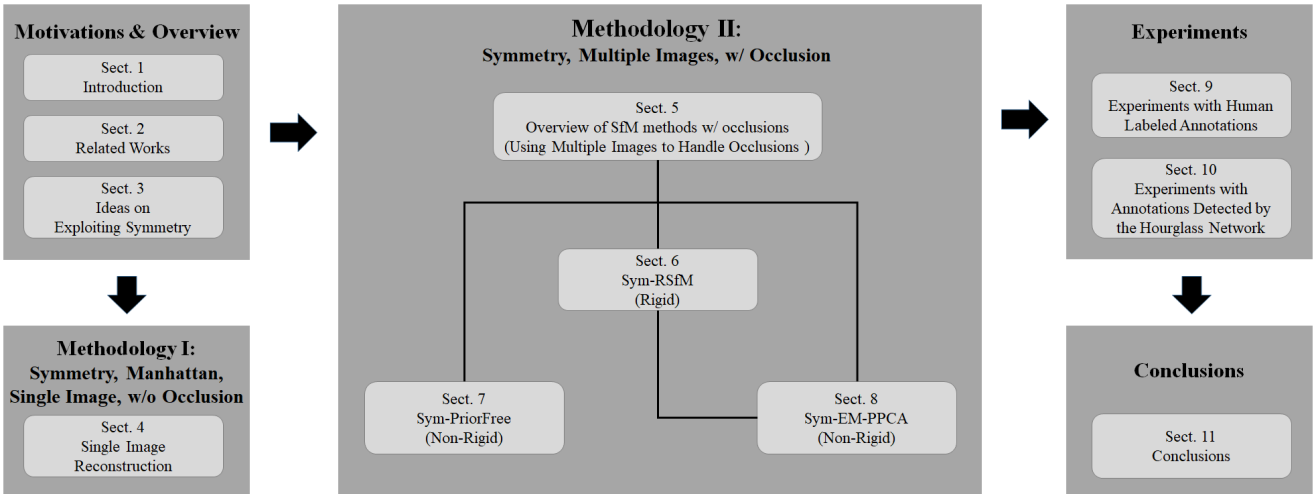

Fig. 2 The overview of this paper. The proposed four methods can be categorized into two parts, *i.e.* , i) the single image reconstruction method which exploits the symmetry and Manhattan properties, this method is based on geometry and does not deal with occlusions; ii) the reconstruction methods which exploits multiple images in an optimization framework, which make use of symmetry property and is able to perform occlusion reasoning. Specifically, we propose three novel methods for multiple-image reconstruction, *i.e.* , Sym-RSfM (Rigid), Sym-PriorFree (Non-Rigid), and Sym-EM-PPCA (Non-Rigid). All of the three methods are related to the energy minimization problem with occluded keypoints introduced in Sect. 5, and Sym-EM-PPCA also uses the results from Sym-RSfM as initialization for further non-rigid updates. As an extension of Sym-RSfM, Sym-PriorFree performs direct factorization with non-rigid deformation, which differs with Sym-EM-PPCA on whether to use a prior to deal with the non-rigid ambiguities.

lines, *i.e.* [12, 13] and [44] respectively, using the projection models specified in the baselines.

Our main contributions are as follows. We show the symmetry can be used, by exploiting a coordinate transformation, to decompose the estimation of the 3D structure into different components which can be addressed independently (subject to complications caused by unobserved keypoints and factorization, or gauge ambiguities). We also show that Manhattan constraints are sufficient to estimate the camera projection, provided that sufficient keypoints can be observed (which is rare in practice). This enables us to specify four new algorithms for estimating 3D structure and camera parameters. Specifically, as shown in Fig. 2, the four novel algorithms are:

- Single image 3D structure reconstruction, as detailed in Sect. 4, which exploits both symmetry and Manhattan properties of a single object.
- *Sym-RSfM* (rigid), which takes multiple images as input and assumes rigid deformations, with symmetry constraints. This method is mainly discussed in Sect. 6, which also makes use of the content introduced in Sect. 5 for occlusion reasoning.
- *Sym-PriorFree* (non-rigid), which performs direct matrix factorization on multiple images with non-rigid deformation and symmetry constraint, to initialize a coordinate descent algorithm. We detail this method in Sect. 7, which also uses Sect. 5 to recover the occlusions.
- *Sym-EM-PPCA* (non-rigid), which imposes symmetric constraints on both the 3D structure and deformation bases. Sym-RSfM is used to initialize Sym-EM-PPCA to impose hard symmetric constraints on the 3D structure. We discuss this method in Sect. 8, where the initialization of this method is obtained by *Sym-RSfM* (*i.e.* , Sect. 5 and Sect. 7).

We provide detailed experiments showing the performance of our algorithms on objects in the Pascal3D+ dataset [47]. In all cases, we compare the performance of our methods, which exploit symmetry, to the baselines which do not. Note that lack of commonality between the tasks, and the baselines (which use a variety of camera projection models), means that it is hard to give precise fair comparisons between all our methods (so we concentrate on their performance relative to the baselines).

Our experiments are done in two settings. Firstly, we use the ground truth 2D annotations specified in Pascal3D+. We note that the annotations only include keypoints which are observable (*i.e.* the annotators did not try to estimate the positions of the occluded keypoints). We also showed the robustness of our methods to location errors in the keypoint locations, by adding Gaussian noise, and we report these results in the supplementary material. Secondly, we applied our method

to real images using a stacked hourglass deep network [36] to estimate the keypoint positions (and identifying their semantic meaning, *e.g.* left front wheel) as input to our algorithms. These results are generally very good, perhaps because the stacked hourglass network outputs good estimates for the positions of the missing (*i.e.* occluded) keypoints.

We note that our algorithms were first presented in two separate conference papers [15,16], which reported results on Pascal3D+. The novelties of this paper are to unify the work in the two papers by presenting them in a way which simplifies them and explains the common ideas they rely on (*e.g.* the change of coordinates, and the decomposition of the 3D structure into different components). In addition, we have added the experiments on real images, using the stacked hourglass deep network, to show that our approach can be applied to the real-world problem.

The rest of the paper is organized as follows. Firstly, we review related work in Sect. 2. Next, Sect. 3 introduces our main ideas for exploiting symmetry without considering missing keypoints. Following this idea, we give details of our single image reconstruction in Sect. 4. We overview our SfM methods for dealing with occlusions in Sect. 5, where we define the *full energy function* (with missing data) and the *surrogate energy function* (without missing data or with missing data initialized). Then, we give details of the three proposed symmetric SfM methods, *i.e.* Sym-RSfM, Sym-PriorFree, and Sym-EM-PPCA, in Sects. 6 - 8, respectively. We note that Sym-PriorFree (non-rigid SfM) is a direct extension of Sym-RSfM (rigid SfM), and Sym-EM-PPCA (non-rigid SfM) is an indirect extension using Sym-RSfM as initialization. Followed this, we perform experiments on Pascal3D+ with manually labeled keypoints and keypoints localized by a stacked hourglass network in Sects. 9 and 10, respectively. Finally, we give our conclusions in Sect. 11.

2 Related Works

Symmetry has been studied in computer vision for several decades. For example, symmetry has been used as a cue in depth recovery [17,27,35] as well as for recognizing symmetric objects [45]. Grossmann and Santos-Victor have utilized various geometric clues, such as planarity, orthogonality, parallelism, and symmetry, for 3D scene reconstruction [20,21], where the camera rotation matrix was precomputed by vanishing points [19]. Recently, researchers have applied symmetry to scene reconstruction [25], and 3D mesh reconstruction with occlusion [41]. In addition, symmetry, incorporated with planarity and a compactness prior, has also been studied to reconstruct structures defined by 3D keypoints [29]. By contrast, the Manhattan world assumption was developed originally for scenes [10, 11, 14], where the authors assumed visual scenes are based on a Manhattan 3D grid which provides 3 perpendicular axis constraints. Both symmetry and Manhattan can be straightforwardly combined, and adapted to 3D object reconstruction, particularly for man-made objects.

The estimation of 3D structure from multiple images is one of the most active research areas in computer vision. Classic structure from motion (SfM) for rigid deformation builds on matrix factorization methods [28, 42], where rigid SfM with missing keypoints was also studied in [33]. Then, more general non-rigid deformation [30–32] was considered, and the rigid SfM in [28, 42] was extended to non-rigid case by Bregler *et al.* [7]. Non-rigid SfM was shown to have ambiguities [48] and various non-rigid SfM methods were proposed using priors on the non-rigid deformations [3, 4, 18, 37, 44, 48]. Gotardo and Martinez proposed a Column Space Fitting (CSF) method for rank-r matrix factorization and applied it to SfM with smooth time-trajectories assumption [18]. A more general framework for rank-r matrix factorization was proposed by Hong and Fitzgibbon [24], which contained the CSF method as a special case[1]. More recently, it has been proved that the ambiguities in non-rigid SfM do not affect the estimated 3D structure, [2] which leads to prior free matrix factorization methods [12, 13].

Researchers have used SfM methods for category-specific object reconstruction, *e.g.* estimating the structure of *cars* from images of different *cars* under various viewing conditions [26, 46], where the data was augmented by symmetry in [26] (*i.e.* left-right flipping), but these did not exploit symmetry or Manhattan in their reconstruction algorithm. We point out that repetition patterns have recently been incorporated into SfM for urban facades reconstruction in [8], but this work focused mainly on repetition detection and registration.

3 Exploiting Symmetry to Estimate 3D Structure

This section specifies the basic set up of our approach and, in particular, how we model and exploit symmetry. In the following sections we will apply these basic ideas

[1] However, the general framework in [24] cannot be used to SfM directly, because it did not constrain that all the keypoints within the same frame should have the same translation. Instead, [24] focused on better optimization of rank-r matrix factorization and better runtime.

to cases where only a single image is available (Sect. 4), multiple images are available with *rigid* deformation among the objects (Sect. 6), and multiple images are available with *non-rigid* deformation among the objects (Sects. 7 and 8). In this paper, objects are represented by keypoints. For simplicity of exposition, this section assumes that all the keypoints of the objects are observed (*i.e.* without occlusion) and we will refer to the following sections for how we deal with missing/occluded keypoints.

In this paper, we group keypoints into keypoint pairs and use a superscript † to denote symmetry, *i.e.* Y and $Y^\dagger$ are the 2D projections of symmetric keypoint pairs, where we use *Italic letter* (*e.g.* Y) to denote a matrix for one image. *Blackboard letter* (*e.g.* $\mathbb{Y}$) is used for vectorizing a matrix (*e.g.* $\mathbb{Y} = \text{vec}(Y)$). *Bold letter* (*e.g.* $\mathbf{Y}$) denote a matrix for stacking multiple images. Finally, we use *Calligraphic letter* (*e.g.* $\mathcal{A}$) to represent matrix (row/column) manipulation operator.

We organize an object consisting of $2P$ keypoints as P keypoint-pairs. Without loss of generality, we assume that the object is symmetric along the x-axis in the *world* coordinates (as we can always rotate the *world* coordinates). A *keypoint pair* consists of two points $S_p = [x_p, y_p, z_p]^\top$ and $S_p^\dagger = [-x_p, y_p, z_p]$. The object is represented by S, $S^\dagger \in \mathbb{R}^{3\times P}$, consisting of the set $\{(S_p,\ S_p^\dagger) : p = 1, ..., P\} \in \mathbb{R}^{3\times 2P}$. S and $S^\dagger$ are related by $S^\dagger = \mathcal{A}S$, where $\mathcal{A}$ is the matrix operator $\mathcal{A} = \text{diag}([-1, 1, 1])$. We will use $S_x = \{x_p : p = 1, ..., P\} \in \mathbb{R}^{1\times P}$ and $S_{yz} = \{(y_p;\ z_p) : p = 1, ..., P\} \in \mathbb{R}^{2\times P}$ to represent the x-components and the y, z-components respectively. Note that knowing S_x and S_{yz} is equivalent to knowing the 3D structure S, $S^\dagger$.

Let Y, $Y^\dagger \in \mathbb{R}^{2\times P}$ be the observed 2D coordinates of all the P symmetric pairs. We assume orthographic projection which implies that:

$$Y = RS, \quad Y^\dagger = RS^\dagger, \tag{1}$$

where $R \in \mathbb{R}^{2\times 3}$ is the camera projection matrix. We eliminate translation by centralizing the 2D keypoints.

In order to exploit symmetry between the keypoint-pairs, we perform a *change of coordinates from* Y, $Y^\dagger$ *to* L, M by:

$$L = \frac{Y - Y^\dagger}{2} \qquad M = \frac{Y + Y^\dagger}{2}. \tag{2}$$

Hence L represents the difference between the projections of keypoint pairs, while M represents the sum of their projections.

This change of coordinates *decouples the dependence of the projections into a term L which depends only on the x-coordinates of the keypoints and a term M which depends only on their y, z coordinates.* More specifically, L depends only on S_x while M depends only on S_{yz}. Moreover, the projection matrix R can be decomposed into two terms $R^1 \in \mathbb{R}^{2\times 1}$ and $R^2 \in \mathbb{R}^{2\times 2}$, which represent the *first column* and *second-third double-column* of R, respectively:

$$R = [R^1, R^2], \tag{3}$$

so that we obtain two separate projections:

$$L = R^1 S_x, \quad M = R^2 S_{yz}. \tag{4}$$

For the single image case, we can use Eq. (4) to solve for S_x and S_{yz} provided the rotation R is known. In Sect. 4 we first describe how the Manhattan assumption can be used to solve for R and then show how S_x and S_{yz} can be computed using Eq. (4).

Next we consider the case of estimating the object structure from multiple views $n = 1, ..., N$. We represent the object by $\mathbf{S} = \{S_n : n = 1, ..., N\} \in \mathbb{R}^{3N\times P}$ and $\mathbf{S}^\dagger = \{S_n^\dagger : n = 1, ..., N\} \in \mathbb{R}^{3N\times P}$, where the symmetry relation $S_n^\dagger = \mathcal{A}S_n$ applies for all n. For objects with rigid deformation, the structure is fixed (*i.e.* all S_n are identical only up to scales and rotations). For objects with non-rigid deformation, we assume that the S_n, $S_n^\dagger$ can be expressed as a linear combination of unknown basis functions. We use the same orthographic projection as Eq. (1) to express the observations $\mathbf{Y}$, $\mathbf{Y}^\dagger \in \mathbb{R}^{2N\times P}$ in terms of the projections $\mathbf{R} = \{R_n : n = 1, ..., N\} \in \mathbb{R}^{2N\times 3N}$ and the 3D structure:

$$\mathbf{Y} = \mathbf{RS}, \quad \mathbf{Y}^\dagger = \mathbf{RS}^\dagger. \tag{5}$$

As before, we perform the change of coordinates and decouple the projections into terms that depend on the x and yz-components of the 3D structure:

$$\mathbf{L} = \frac{\mathbf{Y} - \mathbf{Y}^\dagger}{2} = \mathbf{R}^1\mathbf{S}_x, \quad \mathbf{M} = \frac{\mathbf{Y} + \mathbf{Y}^\dagger}{2} = \mathbf{R}^2\mathbf{S}_{yz}, \tag{6}$$

where $\mathbf{R}^1 = \{R_n^1 : n = 1, ..., N\} \in \mathbb{R}^{2N\times N}$ consists of the first column of every R_n, $\mathbf{R}^2 = \{R_n^2 : n = 1, ..., N\} \in \mathbb{R}^{2N\times 2N}$ consists of the second-third double-column of every R_n, and $R_n = [R_n^1, R_n^2]$.

Following the standard structure from motion formulations, we seek to estimate the 3D structure by using a least squares energy function to minimize:

$$Q(\mathbf{S}, \mathbf{R}) = ||\mathbf{Y} - \mathbf{RS}||_2^2 + ||\mathbf{Y}^\dagger - \mathbf{RS}^\dagger||_2^2. \tag{7}$$

This consists of the sum of two energy terms. Standard factorization techniques, e.g., singular value decomposition (SVD), could be applied to minimize each energy term separately. But this is problematic because the unknowns in both terms are coupled, *i.e.* they contain the

same $\mathbf{R}$ and $\mathbf{S}^\dagger = \mathcal{A}\mathbf{S}$, therefore leading impossibility to minimize each energy term separately.

Instead, we use the change of coordinates and re-express the energy function in terms of two energy terms which are functions of different/decoupled variables following Eq. (6):

$$Q(\mathbf{S}, \mathbf{R}) = ||\mathbf{L} - \mathbf{R}^1\mathbf{S}_x||_2^2 + ||\mathbf{M} - \mathbf{R}^2\mathbf{S}_{yz}||_2^2. \tag{8}$$

For the rigid case, this enables us to use SVD to solve energy term separately to estimate $\mathbf{R}^1, \mathbf{S}_x$ and $\mathbf{R}^2, \mathbf{S}_{yz}$.

For the non-rigid case, it enables us to extend the factorizable prior-free methods of [12, 13]. These methods require that the objects are represented in terms of a linear combination of bases:

$$\mathbf{Y} = \mathbf{R}\mathbf{S} \quad \text{and} \quad \mathbf{S} = \mathbf{V}\mathbf{z}, \quad \mathbf{R}\mathbf{R}^\top = I, \tag{9}$$

where $\mathbf{V}$ are the bases and $\mathbf{z}$ are the coefficients. The bases are fixed, but the coefficients change for different viewpoints.

But the factorization approaches also involve gauge ambiguities which must be identified and addressed. In classic structure from motion, singular value decomposition specifies the solution $\mathbf{R}, \mathbf{S}$ up to an ambiguity $\mathbf{R} \leftarrow \mathbf{R}\mathbf{A}_1$ and $\mathbf{S} \leftarrow \mathbf{A}_1^{-1}\mathbf{S}$ where $\mathbf{A}$ is an invertible matrix. But this ambiguity $\mathbf{A}$ is restricted to be a rotation matrix provided the camera projection is orthogonal, hence it is only a "gauge freedom" [34], corresponding to a choice of coordinate system.

For the non-rigid case, the gauge ambiguities are more complicated. Eq. (9) introduces additional ambiguities of the non-rigid SfM between the deformation bases $\mathbf{V}$ and their coefficients $\mathbf{z}$ [2]. Specifically, let $\mathbf{A}_2$ be another invertible matrix, and let $\mathbf{w}$ lie in the null space of the projected deformation bases $\mathbf{RV}$, then $\mathbf{z} \leftarrow \mathbf{A}_2\mathbf{z}$ and $\mathbf{V} \leftarrow \mathbf{V}\mathbf{A}_2^{-1}$, or $\mathbf{z} \leftarrow \mathbf{z} + \alpha\mathbf{w}$ will not change the value of $\mathbf{RVz}$. This motivated [44] to impose a Gaussian prior on the coefficient $\mathbf{z}$ in order to eliminate the ambiguities. But, as proved in [2] these ambiguities are also gauge freedoms, *i.e.* they do not affect the estimate of the 3D structure. This proof facilitated prior-free matrix factorization methods for non-rigid SfM [12, 13].

Our symmetric formulations alter these gauge ambiguities. For the rigid case, see Sect 6, we identify the ambiguities and specify strategies to deal with them. For the non-rigid case, we specify two algorithms which exploit symmetry. The first one in Sect. 7 is a prior-free method based on [12, 13] which treats them as gauge ambiguities. Sect. 8 describe the second one which follows [44] by using a prior to remove the ambiguities.

There remains, however, the serious limitation that several keypoints will be missing due to occlusion. This requires treating the missing keypoints as hidden variables and determining strategies to initialize and update them. We leave the details of this to Sect. 5.

4 3D Reconstruction from a Single Image

In this section, we describe how to reconstruct the 3D structure of an object from a single image using its symmetry and Manhattan properties. We first show how to exploit symmetry to estimate the 3D structure if the camera projection is known. Then we describe how the camera projection can be estimated by using the Manhattan structure, which is sufficient to determine the camera projection up to sign ambiguities (*e.g.* we cannot distinguish between front-to-back and back-to-front directions). We assume orthographic projection and estimate the 3D structure only for the keypoint pairs which are detected.

4.1 Estimate 3D Structure Exploiting Symmetry with Known Camera Projection

Let $Y, Y^\dagger \in \mathbb{R}^{2\times P}$ be the observed 2D coordinates of all the P symmetric pairs, then orthographic projection implies $Y = RS, \quad Y^\dagger = RS^\dagger$. We represent the camera projection by $R = [R^1, R^2]$, where:

$$R^1 = \begin{bmatrix} r_{11} \\ r_{21} \end{bmatrix}, \quad R^2 = \begin{bmatrix} r_{12}, r_{13} \\ r_{22}, r_{23} \end{bmatrix}. \tag{10}$$

We change the coordinates to $L = \frac{Y-Y^\dagger}{2}, M = \frac{Y+Y^\dagger}{2}$ and obtain:

$$L = \begin{bmatrix} r_{11}x_1, ..., r_{11}x_P \\ r_{21}x_1, ..., r_{21}x_P \end{bmatrix}, \tag{11}$$

$$M = \begin{bmatrix} r_{12}y_1 + r_{13}z_1, ..., r_{12}y_P + r_{13}z_P \\ r_{22}y_1 + r_{23}z_1, ..., r_{22}y_P + r_{23}z_P \end{bmatrix}. \tag{12}$$

We re-express this as:

$$L = R^1 S_x, \quad M = R^2 S_{yz}, \tag{13}$$

where $S_x = [x_1, ..., x_P]$ and $S_{yz} = \begin{bmatrix} y_1, ..., y_P \\ z_1, ..., z_P \end{bmatrix}$.

If the rotation matrix R is known, then we can solve Eqs. (11) and (12) to estimate the components (x_p, y_p, z_p) of all the points S_p and hence, recover the 3D structure. Observe that we have only just enough equations to solve for the (y_p, z_p) uniquely. On the other hand, x_p is over-determined due to symmetry. We also note that the problem (*i.e.* recovering the 3D structure with known projection R) is ill-posed if we do not exploit symmetry, *i.e.* it involves inverting a 2×3 projection matrix.

4.2 Estimate Camera Projection with Manhattan Property

This section shows that using Manhattan property alone is sufficient to recover the projection matrix under an orthographic camera.

The Manhattan assumption is that *the objects in the image possess a natural Cartesian coordinate system where three perpendicular orientations can be inferred* [11]. We say that the object which satisfies the Manhattan assumption has Manhattan properties. This commonly exists in man-made objects. An example is shown in Fig. 1 (Right), where left wing → right wing, nose → tail, and top rudder → bottom rudder are three Manhattan directions. In mathematical terms, the Manhattan assumptions means that we can identify six keypoints $S_a, S_b, S_c, S_d, S_e, S_f$ such that the three vectors $S_a - S_b, S_c - S_d, S_e - S_f$ are orthogonal and point along the main axes of the object. Intuitively, Manhattan properties allow us to extract 3D information from 2D images. In the following, we will show mathematically that the Manhattan properties can be used for estimating the viewpoint of a 2D image under an orthographic camera.

Consider a single Manhattan axis specified by 3D points S_a and S_b. Without loss of generality, assume that these points are along the x-axis, *i.e.* $S_a - S_b = [\Delta x, 0, 0]^\top$, where Δx is the distance between the keypoints. It follows from the projection that:

$$Y_a - Y_b = R(S_a - S_b) = \begin{bmatrix} r_{11}\Delta x \\ r_{21}\Delta x \end{bmatrix}. \tag{14}$$

Eliminating Δx by division, yields the constraint:

$$r_{11}/r_{21} = (y_a^1 - y_b^1)/(y_a^2 - y_b^2), \tag{15}$$

where (y_a^1, y_a^2) and (y_b^1, y_b^2) are the components of Y_a and Y_b, respectively. We obtain similar constraints by using Manhattan axes in the y, z-axes, *e.g.* assume S_c, S_d are along the y-axis and S_e, S_f are along the z-axis. Defining $\mu_1 = r_{11}/r_{21}$, $\mu_2 = r_{12}/r_{22}$, $\mu_3 = r_{13}/r_{23}$ yields:

$$\begin{aligned} \mu_1 &= r_{11}/r_{21} = (y_a^1 - y_b^1)/(y_a^2 - y_b^2), \\ \mu_2 &= r_{12}/r_{22} = (y_c^1 - y_d^1)/(y_c^2 - y_d^2), \\ \mu_3 &= r_{13}/r_{23} = (y_e^1 - y_f^1)/(y_e^2 - y_f^2). \end{aligned} \tag{16}$$

Now we recall that the orthogonality of R, *i.e.* $RR^\top = I$, which implies:

$$\begin{aligned} &r_{11}^2 + r_{12}^2 + r_{13}^2 = 1, \quad r_{21}^2 + r_{22}^2 + r_{23}^2 = 1, \\ &r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23} = 0. \end{aligned} \tag{17}$$

Substituting r_{11}, r_{12}, r_{13} using Eq. (16) gives the following linear equations for r_{21}, r_{22}, r_{23}:

$$\begin{bmatrix} 1, & 1, & 1 \\ \mu_1^2, & \mu_2^2, & \mu_3^2 \\ \mu_1, & \mu_2, & \mu_3 \end{bmatrix} \begin{bmatrix} r_{21}^2 \\ r_{22}^2 \\ r_{23}^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \tag{18}$$

These equations can be solved for the unknowns r_{21}^2, r_{22}^2 and r_{23}^2 provided the coefficient matrix (above) is invertible (*i.e.* has full rank). This requires that $(\mu_1 - \mu_2)(\mu_2 - \mu_3)(\mu_3 - \mu_1) \neq 0$. Because μ_1, μ_2, μ_3 are the directions (more precisely, the slopes) of the projected Manhattan axes in 2D space, this prerequisite is violated only in a very special case when the camera optical axis and two Manhattan directions are co-planar (which projects the two Manhattan directions into one same line).

Note that there are sign ambiguities for solving r_{21} r_{22}, r_{23} from $r_{21}^2, r_{22}^2, r_{23}^2$. But these ambiguities do not affect the estimation of the 3D shape, because they are just choices of the coordinate system. Next we can calculate r_{11}, r_{12}, r_{13} directly based on r_{21}, r_{22}, r_{23} and μ_1, μ_2, μ_3. This recovers the camera projection matrix and hence, by exploiting symmetry, we can also estimate the 3D structure from a single image.

5 Overview of Structure from Motion with Missing Keypoints

This section overviews how to jointly estimate the 3D structure, the viewpoint, and recover the occluded 2D keypoints from multiple images by SfM. This is not straightforward because applying factorization to Eq. (8) is not feasible if missing data exists in $\mathbf{L}$ or $\mathbf{M}$. We first formulate the problem in terms of energy minimization with symmetry constraints and missing keypoints in Sect. 5.1. Then, we use coordinate descent, or hard EM if the missing keypoints are treated as latent variables, to optimize the energy function shown in Sect. 5.2. After that, Sect. 5.3 gives a simple method to initialize the occluded keypoints for the coordinate descent/hard EM algorithm. The initialization of the 3D structure and the viewpoint are much more complex, depending on how we assume and how we deal with the ambiguities (as discussed in Sect. 3). We discuss the initialization of the 3D structure and the viewpoint in later sections in Sects. 6 and 7 because there are technical differences between the rigid and non-rigid cases.

5.1 Problem Formulation

If missing keypoints exist, then estimating 3D structure is more complex than described in Sect. 3. This is because applying factorization to Eq. (7), *i.e.* $Q(\mathbf{S}, \mathbf{R}) =$

$||\mathbf{L}-\mathbf{R}^1\mathbf{S}_x||_2^2+||\mathbf{M}-\mathbf{R}^2\mathbf{S}_{yz}||_2^2$, or Eq. (8), *i.e.* $Q(\mathbf{S},\mathbf{R}) = ||\mathbf{Y}-\mathbf{R}\mathbf{S}||_2^2+||\mathbf{Y}^\dagger-\mathbf{R}\mathbf{S}^\dagger||_2^2$, will not be feasible. Hence, we formulate an full energy function with missing points which we then seek to minimize.

To deal with unobserved keypoints we divide them into a *visible set* VS, $VS^\dagger$and an *invisible set* IVS, $IVS^\dagger$. Then the *full energy function* can be formulated as:

$$\begin{aligned}&\mathcal{Q}(R_n, S_n, \{Y_{n,p}, (n,p)\in IVS\}, \{Y^\dagger_{n,p}, (n,p)\in IVS^\dagger\})\\ =&\sum_{(n,p)\in VS}||Y_{n,p}-R_nS_{n,p}||_2^2+\sum_{(n,p)\in VS^\dagger}||Y^\dagger_{n,p}-R_nS^\dagger_{n,p}||_2^2+\\&\sum_{(n,p)\in IVS}||Y_{n,p}-R_nS_{n,p}||_2^2+\sum_{(n,p)\in IVS^\dagger}||Y^\dagger_{n,p}-R_nS^\dagger_{n,p}||_2^2,\end{aligned}\tag{19}$$

where $\{Y_{n,p}, (n,p)\in IVS\}$, $\{Y^\dagger_{n,p}, (n,p)\in IVS^\dagger\}$ are the missing keypoints.

5.2 Optimization of The Full Energy with Occluded Keypoints

We adopt a coordinate descent method to jointly update the unknowns $(R_n, S_n, \{Y_{n,p}, (n,p)\in IVS\}, \{Y^\dagger_{n,p}, (n,p)\in IVS^\dagger\})$. This is equivalent to a hard EM algorithm if we treat the missing keypoints $\{Y_{n,p}, (n,p)\in IVS\}, \{Y^\dagger_{n,p}, (n,p)\in IVS^\dagger\}$ as latent variables, and R_n, S_n as unknown parameters. Note that the energy in Eq. (19) *w.r.t* R_n, S, *i.e.* when the missing points are known (or estimated), can be obtained as:

$$\mathcal{Q}(R_n,S_n)=\sum_n||Y_n-R_nS_n||_2^2+\sum_n||Y^\dagger_n-R_nS^\dagger_n||_2^2,\tag{20}$$

we call this the *surrogate energy function*.

The 3D structure S_n can be updated by optimizing the *surrogate energy function* in Eq. (20):

$$S_n=(R_n^\top R_n+\mathcal{A}^\top R_n^\top R_n\mathcal{A})^{-1}(R_n^\top Y_n+\mathcal{A}^T R_n^T Y_n^\dagger).\tag{21}$$

Note that if rigid deformation is assumed, S_n for different image n will be identical, *i.e.* $S_n=S, \forall n$. Therefore, the common 3D structure S can be updated in its vectorized form $\mathbb{S}$ by:

$$\mathbb{S}=\left(\sum_{n=1}^N(G_n^\top G_n+\mathcal{A}_P^\top G_n^\top G_n\mathcal{A}_P)\right)^{-1}\left(\sum_{n=1}^N(G_n^\top \mathbb{Y}_n+\mathcal{A}_P^\top G_n^\top \mathbb{Y}_n^\dagger)\right).\tag{22}$$

where $\mathbb{S}\in\mathbb{R}^{3P\times 1}, \mathbb{Y}_n\in\mathbb{R}^{2P\times 1}, \mathbb{Y}_n^\dagger\in\mathbb{R}^{2P\times 1}$ are vectorized $S, Y_n, Y_n^\dagger$, respectively. $G_n=I_P\otimes R_n$ and $\mathcal{A}_P=I_P\otimes\mathcal{A}$. $I_P\in\mathbb{R}^{P\times P}$ is an identity matrix.

Each R_n is updated under the orthogonality constraints $R_nR_n^\top=I$ similar to the idea in EM-PPCA [44]: we first parameterize R_n to a full 3×3 rotation matrix Q and update Q by its rotation increment. Please refer to the supplementary material for the details.

Algorithm 1: Optimization of the full energy in Eq. (19) with occluded keypoints.

Input: The stacked keypoint sets (for all the N images) $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ with occluded points, in which each occluded point is set to $\mathbf{0}$ initially.
Output: The camera projection matrix R_n for each image, the common 3D structure S (rigid case) or each S_n (non-rigid case), and the keypoints with recovered occlusions $(\mathbf{Y})^t$ and $(\mathbf{Y}^\dagger)^t$.

```
Initialize the occluded points by Algorithm 2.
Initialize each R_n, the common S or each S_n by Algorithm 3 (rigid case) or Algorithm 4 (non-rigid case), respectively.
repeat
    Update the common S by Eq. (22) (rigid case) or each S_n Eq. (21) (non-rigid case).
    Update each R_n according to the supplementary material.
    Calculate the occluded points by Eq. (24), and update them in Y_n, Y_n^†.
    Centralize the Y_n, Y_n^† by Eq. (25).
until Eq. (19) converge;
```

Using Eq. (19) we can estimate the positions of the occluded points of $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ (*i.e.* the p-th point $Y_{n,p}$ and $Y^\dagger_{n,p}$) by minimizing the following energy:

$$\begin{aligned}\mathcal{Q}(Y_{n,p},Y^\dagger_{n,p})=&\sum_{(n,p)\in IVS}||Y_{n,p}-R_nS_p||_2^2+\\&\sum_{(n,p)\in IVS^\dagger}||Y^\dagger_{n,p}-R_n\mathcal{A}S_p||_2^2.\end{aligned}\tag{23}$$

This gives an update rule which specifies the missing point:

$$Y_{n,p}=R_nS_p,\qquad Y^\dagger_{n,p}=R_n\mathcal{A}S_p,\tag{24}$$

where $(n,p)\in IVS$.

Note that we do not model the translation explicitly as the translation can be assumed being eliminated by centralizing the data. However, since the occluded points have been updated in our method, we have to re-estimate the translation and re-centralize the data. This is done by:

$$\begin{aligned}&Y_n\leftarrow Y_n-\mathbf{1}_P^\top\otimes t_n,\qquad Y_n^\dagger\leftarrow Y_n^\dagger-\mathbf{1}_P^\top\otimes t_n,\\&t_n=\sum_p(Y_{n,p}-R_nS_p+Y^\dagger_{n,p}-R_n\mathcal{A}S_p).\end{aligned}\tag{25}$$

This coordinate descent/hard EM algorithm is not guaranteed to converge to the global optimal, and requires good initialization to obtain good performance.

A simple initialization method for the occluded keypoints $\{Y_{n,p}, (n,p) \in IVS\}, \{Y^{\dagger}_{n,p}, (n,p) \in IVS^{\dagger}\}$ is given in the following subsection.

But the initialization for the 3D structure S_n and the rotation matrix R_n is much more complex, and also differs with rigid or non-rigid deformation assumptions (because of different ambiguities in the factorization methods). These are done by minimizing the surrogate energy function with the missing variables initialized. We will discuss this in detail in Sects. 6 and 7 for the rigid and non-rigid case, respectively. The algorithm to optimize the full energy Eq. (19) is summarized in Algorithm 1.

5.3 Initialization of the Missing Data

In this section, we describe how we initialize the missing data without exploiting the symmetry. This uses all the observed keypoints and can be used to initialize the missing keypoints for the surrogate energy function, or alternatively directly to initialize the full energy.

We use a very simple missing data initialization algorithm, which is similar to Tomasi-Kanade factorization [42] with rank 3 pruning/recovery. This is because: (I) The missing keypoints can be further updated iteratively when we optimize the full energy function. This resembles a general EM algorithm where the missing keypoints are treated as latent variables and updated during the E-step. (II) Missing data recovery is not the main focus of our paper, and we found that this simple algorithm can already yield a good initialization. We point out that more advanced missing data initialization methods exist, such as [33] which can better address the possible degenerate images/frames. We leave incorporating more advanced missing data initializations (such as [33]) as our future work.

Let $\mathbf{Y} = [Y_1^{\top}, ..., Y_N^{\top}]^{\top} \in \mathbb{R}^{2N \times P}, \mathbf{Y}^{\dagger} = [(Y_1^{\dagger})^{\top}, ..., (Y_N^{\dagger})^{\top}]^{\top} \in \mathbb{R}^{2N \times P}$ are the stacked keypoints for all the images, and $\mathbf{R} = [R_1^{\top}, ...R_N^{\top}]^{\top} \in \mathbb{R}^{2N \times 3}$ are the stacked camera projection. Thus, we have $\mathbf{Y}^{\text{All}} = [\mathbf{Y}, \mathbf{Y}^{\dagger}] = \mathbf{R}[S, \mathcal{A}S]$. It implies that $\mathbf{Y}^{\text{All}}$ has the same rank, namely 3, with $\mathbf{R}[S, \mathcal{A}S]$ given all the points of $[S, \mathcal{A}S]$ do not lie on a plane or line. Therefore, rank 3 recovery can be used to initialize the missing points. Also, the same centralization as in the previous section has to be done after each iteration of the missing points.

The initialization of the missing points is summarized in Algorithm 2.

Algorithm 2: The initialization of the occluded points.

Input: The stacked keypoint sets (for all the N images) $\mathbf{Y}$ and $\mathbf{Y}^{\dagger}$ with occluded points, in which each occluded point is set to $\mathbf{0}$ initially. The number of iterations T (default 10).
Output: The keypoints with initially recovered occlusions $(\mathbf{Y})^t$ and $(\mathbf{Y}^{\dagger})^t$.

Set $t = 0$, initialize the occluded points ignoring symmetry by:
while $t < T$ **do**
  Centralize $\mathbf{Y}^{\text{All}} = [(\mathbf{Y})^t, (\mathbf{Y}^{\dagger})^t]$ by Eq. (25).
  Do SVD on $\mathbf{Y}^{\text{All}}$ ignoring the symmetry, *i.e.* $[\mathbf{A}, \Sigma, \mathbf{B}] = \text{SVD}\left(\mathbf{Y}^{\text{All}}\right)$.
  Use the first 3 component of Σ to reconstruct the keypoints $(\mathbf{Y}^{\text{All}})^{\text{new}}$.
  Replace the occluded points in $(\mathbf{Y})^t$, $(\mathbf{Y}^{\dagger})^t$ by these in $(\mathbf{Y}^{\text{All}})^{\text{new}}$ and set $t \leftarrow t + 1$.
end

6 Symmetric Rigid Structure from Motion

This section describes the symmetric rigid structure from motion (Sym-RSfM) method. Note that the problem described here assumes all the keypoints have been initialized. This section relates to Sect. 5 (specifically, Step 2 of Algorithm 1).

By assuming all keypoints are initialized (or estimated), we obtain a matrix factorization problem in Sect. 6.1 of the form described in Sect. 3. To solve this requires analyzing the novel ambiguities which arise when using the symmetry constraints in Sect. 6.2. Finally, in Sect. 6.3, we estimate the 3D structure and camera parameters by solving these ambiguities.

For consistency with our baseline methods [42], we assume orthographic projection and the keypoints are centralized without translation.

6.1 Problem Formulation

Our problem on the rigid SfM with symmetry constraints is:

$$Q(\mathbf{S}, \mathbf{R}) = ||\mathbf{Y} - \mathbf{RS}||_2^2 + ||\mathbf{Y}^{\dagger} - \mathbf{RS}^{\dagger}||_2^2. \tag{26}$$

Note that the energy function in Eq. (26) cannot be solved directly, because the two energy terms are dependent. Therefore, we follow the ideas in Sect. 3 to change the coordinates from $\mathbf{Y}$, $\mathbf{Y}^{\dagger}$ to $\mathbf{L}$, $\mathbf{M}$ to exploit the symmetry, and obtaining the equations:

$$\mathbf{L} = \frac{\mathbf{Y} - \mathbf{Y}^{\dagger}}{2} = \mathbf{R}^1 S_x, \quad \mathbf{M} = \frac{\mathbf{Y} - \mathbf{Y}^{\dagger}}{2} = \mathbf{R}^2 S_{yz}. \tag{27}$$

This re-expresses the energy function in Eq. (26) in the form:

$$Q(\mathbf{R}, S) = ||\mathbf{L} - \mathbf{R}^1 S_x||_2^2 + ||\mathbf{M} - \mathbf{R}^2 S_{yz}||_2^2. \tag{28}$$

6.2 The Ambiguities in Symmetric Rigid Structure from Motion

We have decomposed the energy into two energy terms which can both be solved independently by SVD. For each energy term, there will be an ambiguity in the solution, but we can resolve these by requiring consistency between the solutions of each energy term.

Equation (28) implies that we can estimate $\mathbf{R}^1, S_x$ and $\mathbf{R}^2, S_{yz}$ by matrix factorization on $\mathbf{L}$ and $\mathbf{M}$ independently up to ambiguities. Then we combine them to remove this ambiguity by exploiting the orthogonality constraints on each R_n: *i.e.* $R_n R_n^\top = I$. Applying SVD to $\mathbf{L}$ and $\mathbf{M}$ gives us estimates, $\hat{\mathbf{R}}^1, \hat{S}_x, \hat{\mathbf{R}}^2, \hat{S}_{yz}$, of $\mathbf{R}^1, S_x$ and $\mathbf{R}^2, S_{yz}$ up to ambiguities λ and B:

$$\begin{aligned}\mathbf{L} &= \mathbf{R}^1 S_x = \hat{\mathbf{R}}^1 \lambda \lambda^{-1} \hat{S}_x, \\ \mathbf{M} &= \mathbf{R}^2 S_{yz} = \hat{\mathbf{R}}^2 B B^{-1} \hat{S}_{yz}.\end{aligned} \tag{29}$$

Here $\mathbf{R}^1$ and $\mathbf{R}^2$ are the decomposition of the true projection matrix $\mathbf{R}$, *i.e.* $\mathbf{R} = [\mathbf{R}^1, \mathbf{R}^2]$, and $\hat{\mathbf{R}}^1$ and $\hat{\mathbf{R}}^2$ are the output from SVD. Equations (29) shows that there is a scale ambiguity λ between $\hat{\mathbf{R}}^1$ and $\mathbf{R}^1$, and a 2-by-2 matrix ambiguity $B \in \mathbb{R}^{2\times 2}$ between $\hat{\mathbf{R}}^2$ and $\mathbf{R}^2$.

Observe from Eq. (29) that the ambiguities (*i.e.* λ and B) are the same for the projection matrices of all the images. For the following derivation, we analyze the ambiguity for the n'th image, *i.e.* projection matrix R_n. Using Eqs. (29), we represent the true projection R_n by:

$$R_n = [R_n^1, R_n^2] = [\hat{R}_n^1, \hat{R}_n^2] \begin{bmatrix} \lambda, & \mathbf{0} \\ \mathbf{0}, & B \end{bmatrix} = \hat{R}_n \begin{bmatrix} \lambda, & \mathbf{0} \\ \mathbf{0}, & B \end{bmatrix}, \tag{30}$$

where $R_n^1 \in \mathbb{R}^{2\times 1}$ and $R_n^2 \in \mathbb{R}^{2\times 2}$ are the first single column and second-third double columns of the true projection matrix R_n. $\hat{R}_n^1 \in \mathbb{R}^{2\times 1}$ and $\hat{R}_n^2 \in \mathbb{R}^{2\times 2}$ are the estimation of R_n^1 and R_n^2 from the matrix factorization, and $\hat{R}_n = [\hat{R}_n^1, \hat{R}_n^2]$.

The main idea to solve the symmetry-induced ambiguities λ and B is to exploit the *orthogonality constraint*, namely

$$R_n R_n^\top = I. \tag{31}$$

Remark 1 Equations (30) and (31) imply that the ambiguity between $\mathbf{R}^1$ and $\hat{\mathbf{R}}^1$, (*i.e.* in the symmetry direction), is just a sign change, which is caused by calculating λ from λ^2. In other words, the symmetry direction can be fixed to be the x-axis in our coordinate system using the decomposition Eq. (27). We have a 2×2 rotation ambiguity between $\mathbf{R}^2$ and $\hat{\mathbf{R}}^2$, by calculating B from $BB^\top$.

Algorithm 3: Symmetric Rigid Structure from Motion (Without Occlusions).

Input: The stacked keypoint sets $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ from N images (without occlusions).
Output: The common 3D structure S, and the camera matrix R_n for each image.

1. Change the coordinates to decouple the symmetry constraints by Eq. (27).
2. Get $\hat{\mathbf{R}}^1, \hat{\mathbf{R}}^2, \hat{S}_x, \hat{S}_{yz}$ by doing SVD on $\mathbf{L}, \mathbf{M}$, *i.e.* Eq. (29).
3. Solve the squared ambiguities λ^2 and $BB^\top$ by Eq. (35).
4. Solve for λ from λ^2, and B from $BB^\top$, up to sign and rotation ambiguities.
5. Recover $\mathbf{R}$ and S by Eq. (36).

6.3 Solve the Ambiguities to Estimate the True 3D Structure and Rotation Matrix

In order to solve the ambiguities, we expend $\hat{R}_n$ by:

$$\hat{R}_n = [\hat{R}_n^1, \hat{R}_n^2] = \begin{bmatrix} \hat{r}_n^{1,1}, & \hat{r}_n^{1,2:3} \\ \hat{r}_n^{2,1}, & \hat{r}_n^{2,2:3} \end{bmatrix}. \tag{32}$$

Substituting Eqs. (30) and (32) into Eq. (31) (by derivations detailed in the supplementary material) yields:

$$\begin{aligned} &A_n \mathbf{x} = [1\ 1\ 0]^\top, \\ &A_n = \begin{bmatrix} (\hat{r}_n^{1,1})^2, & \hat{r}_n^{1,2:3} \otimes \hat{r}_n^{1,2:3} \\ (\hat{r}_n^{2,1})^2, & \hat{r}_n^{2,2:3} \otimes \hat{r}_n^{2,2:3} \\ \hat{r}_n^{1,1}\hat{r}_n^{2,1}, & \hat{r}_n^{1,2:3} \otimes \hat{r}_n^{2,2:3} \end{bmatrix} \begin{bmatrix} 1\ 0\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0\ 0 \\ 0\ 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 0\ 1 \end{bmatrix}^\top \in \mathbb{R}^{3\times 4}, \\ &\mathbf{x} = [\lambda^2, bb_1, bb_2, bb_3]^\top \in \mathbb{R}^{4\times 1}, \end{aligned} \tag{33}$$

where bb_1, bb_2, bb_3 come from $BB^\top$, *i.e.* $BB^\top = \begin{bmatrix} bb_1 & bb_2 \\ bb_2 & bb_3 \end{bmatrix}$, and $\otimes$ denotes Kronecker product.

Stacking A_n for all the images gives a set of over-determined equations for the unknown $\mathbf{x}$ (*i.e.* $3N$ equations for 4 unknowns):

$$\begin{aligned} &\mathbf{A}\mathbf{x} = \mathbf{b}, \\ &\mathbf{A} = [A_1^\top, ..., A_N^\top]^\top \in \mathbb{R}^{3N\times 4}, \\ &\mathbf{b} = \mathbf{1}_N \otimes [1, 1, 0]^\top \in \mathbb{R}^{3N\times 1}. \end{aligned} \tag{34}$$

These over-determined linear equations can be solved efficiently by LSE:

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}. \tag{35}$$

Equation (35) gives the estimates of λ^2 and $BB^\top$. Recovering λ from λ^2 is straightforward up to a sign ambiguity. We also show in the supplementary material that B can be recovered up to a rotation ambiguity on the yz-plane, which also does not affect the reconstructed 3D structure.

Given $\lambda, B, \hat{\mathbf{R}}, \hat{S}$, we get the initial estimation of the true $\mathbf{R}$ and S by:

$$\mathbf{R} = \hat{\mathbf{R}} \begin{bmatrix} \lambda, & \mathbf{0} \\ \mathbf{0}, & B \end{bmatrix}, \qquad S = \begin{bmatrix} \lambda, & \mathbf{0} \\ \mathbf{0}, & B \end{bmatrix}^{-1} \hat{S}. \tag{36}$$

The algorithm for Sym-RSfM (without occlusions) is summarized in Algorithm 3. The full Sym-RSfM method with occlusion reasoning is summarized in Fig. 3, where the algorithm discussed in this section, *i.e.* , Algorithm 3, is used to initialize the projection energy minimization (with occlusions) problem in Algorithm 1.

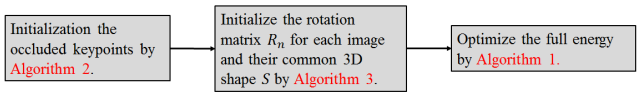

Fig. 3 The flowchart of the full rigid Sym-RSfM method with occlusion reasoning.

7 The Symmetric Prior-Free Matrix Factorization Method for Non-Rigid Structure from Motion

This section, and the following section, addresses how to exploit the symmetry constraints for multiple image 3D structure estimation when the deformation is non-rigid.

Recall that the ambiguities in non-rigid SfM are more complex than those in rigid SfM, this is due to the combination of the deformation bases and the coefficient for non-rigid deformation. As mentioned earlier, the non-rigid SfM methods differ in ways how they deal with this additional ambiguities.

In this section, we rely on the recent research that *these additional ambiguities are also gauge freedoms, which do not affect the estimate of the 3D structure* [2, 12, 13] to develop our non-rigid SfM method exploiting symmetry. These pioneer research in [2, 12, 13] enables us to achieve a direct matrix factorization method for non-rigid SfM which exploits symmetry, which is named Sym-PriorFree method.

Sym-PriorFree is very similar to, and can be regarded as a direct extension of, Sym-RSfM. Similarly, we first give the problem formulation in Sect. 7.1. Then, the ambiguities under symmetry constraints are analyzed in Sect. 7.2, and we solve these ambiguities in Sect. 7.3. Finally, in Sect. 7.4, we discuss how to estimate the 3D structure and the rotation matrix after solving the ambiguities, because this is more complicated than for the rigid case.

Also note that the presentation in this section assumes that all keypoints are either visible or have been initialized/estimated. We deal with missing keypoints using similar methods to those used for the rigid case. More specifically, we can also simultaneously recover them by Algorithm 1 (Step 2) by the Sym-PriorFree method described here as initialization.

7.1 Problem Formulation

Our derivation follow the strategy for exploiting symmetry outlined in Sect. 3. Assume that $Y_n \in \mathbb{R}^{2\times P}$ and $Y_n^\dagger \in \mathbb{R}^{2\times P}$ are the P keypoint pairs for image n without occlusions, we have:

$$\begin{aligned} Y_n &= R_n S_n = [z_{n1}R_n, ..., z_{nK}R_n][\mathbf{V}_1, ..., \mathbf{V}_K]^\top = \mathit{\Pi}_n \mathbf{V}, \\ Y_n^\dagger &= R_n S_n^\dagger = [z_{n1}R_n, ..., z_{nK}R_n][\mathbf{V}_1^\dagger, ..., \mathbf{V}_K^\dagger]^\top = \mathit{\Pi}_n \mathbf{V}^\dagger, \end{aligned} \tag{37}$$

where $\mathbf{z}_n = [z_{n1}, ..., z_{nK}] \in \mathbb{R}^{1\times K}$, $\mathit{\Pi}_n = R_n(\mathbf{z}_n \otimes I_3) \in \mathbb{R}^{2\times 3K}$, and $\mathbf{V} = [\mathbf{V}_1^\top, ..., \mathbf{V}_K^\top]^\top \in \mathbb{R}^{3K\times P}$.

Let $\mathbf{Y}$ be the stacked keypoints of N images, $\mathbf{Y} = [Y_1^\top, ..., Y_N^\top]^\top \in \mathbb{R}^{2N\times P}$, the model is represented by:

$$\begin{aligned} \mathbf{Y} =& \mathbf{RS} = \begin{bmatrix} R_1 S_1 \\ \vdots \\ R_N S_N \end{bmatrix} = \begin{bmatrix} z_{11}R_1, & ..., & z_{1K}R_1 \\ \vdots & \ddots & \vdots \\ z_{N1}R_N, & ..., & z_{NK}R_N \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_K \end{bmatrix} \\ =& \mathbf{\Pi V}, \end{aligned} \tag{38}$$

where $\mathbf{R} = \text{blkdiag}([R_1, ..., R_N]) \in \mathbb{R}^{2N\times 3N}$ are the stacked camera projection matrices, in which blkdiag denotes block diagonal. $\mathbf{S} = [S_1^\top, ..., S_N^\top]^\top \in \mathbb{R}^{3N\times P}$ are the stacked 3D structures. $\mathbf{\Pi} = \mathbf{R}(\mathbf{z} \otimes I_3) \in \mathbb{R}^{2N\times 3K}$, where $\mathbf{z} \in \mathbb{R}^{N\times K}$ are the stacked coefficients. Similar equations apply to $\mathbf{Y}^\dagger$.

Note that $\mathbf{R} \in \mathbb{R}^{2N\times 3N}$, $\mathbf{V} \in \mathbb{R}^{3K\times P}$ are stacked differently from how they were stacked for the Sym-SfM method (*i.e.* $\mathbf{R} \in \mathbb{R}^{2N\times 3}$, $\mathbf{V} \in \mathbb{R}^{3P\times K}$). It is because now we have N different S_n's (*i.e.* $\mathbf{S} \in \mathbb{R}^{3N\times P}$), while there is only one common S in the Sym-RSfM method.

In the following, we assume the deformation bases are symmetric, which ensures that the non-rigid structures are symmetric (*e.g.* the deformation from *sedan* to *truck* is non-rigid and symmetric since *sedan* and *truck* are both symmetric). This yields an energy function:

$$\begin{aligned} \mathcal{Q}(\mathbf{R}, \mathbf{S}) =& ||\mathbf{Y} - \mathbf{RS}||_2^2 + ||\mathbf{Y}^\dagger - \mathbf{RS}^\dagger||_2^2 \\ =& ||\mathbf{Y} - \mathbf{\Pi V}||_2^2 + ||\mathbf{Y}^\dagger - \mathbf{\Pi V}^\dagger||_2^2. \end{aligned} \tag{39}$$

Remark 2 Note that we cannot use the first equation of Eq. (39) to solve $\mathbf{R}, \mathbf{S}$ directly (even if not exploiting symmetry), because $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ are of rank $min\{2N, 3K, P\}$ but estimating $\mathbf{R}, \mathbf{S}$ directly by SVD on $\mathbf{Y}$ and/or

$\mathbf{Y}^\dagger$ requires rank $3N$ matrix factorization. Hence we focus on the last equation of Eq. (39) to get the initialization of $\mathbf{\Pi}, \mathbf{V}$ firstly. Then, $\mathbf{R}, \mathbf{S}$ can be updated by coordinate descent on the first equation of Eq. (39) under *orthogonality constraints* on $\mathbf{R}$ and *low-rank* constraint on $\mathbf{S}$.

We perform the coordinate transformation from $\mathbf{Y}$, $\mathbf{Y}^\dagger$ to $\mathbf{L}$, $\mathbf{M}$ as Sect. 3:

$$\mathbf{L} = \frac{\mathbf{Y} - \mathbf{Y}^\dagger}{2} = \hat{\mathbf{\Pi}}^1 \hat{\mathbf{V}}_x, \qquad \mathbf{M} = \frac{\mathbf{Y} + \mathbf{Y}^\dagger}{2} = \hat{\mathbf{\Pi}}^2 \hat{\mathbf{V}}_{yz}, \tag{40}$$

where $\hat{\mathbf{\Pi}}^1 \in \mathbf{R}^{2N \times K}$ and $\hat{\mathbf{\Pi}}^2 \in \mathbf{R}^{2N \times 2K}$, $\hat{\mathbf{V}}_x \in \mathbf{R}^{K \times P}$ and $\hat{\mathbf{V}}_{yz} \in \mathbf{R}^{2K \times P}$ are both independent.

This yield two independent energies to be minimized separately by SVD:

$$\mathcal{Q}(\mathbf{\Pi}, \mathbf{V}) = ||\mathbf{L} - \hat{\mathbf{\Pi}}^1 \hat{\mathbf{V}}_x||_2^2 + ||\mathbf{M} - \hat{\mathbf{\Pi}}^2 \hat{\mathbf{V}}_{yz}||_2^2. \tag{41}$$

7.2 The Ambiguities in Symmetry Prior Free Matrix Factorization Method

Solving Eq. (41) by matrix factorization gives us solutions up to a matrix ambiguity H. More precisely, there are ambiguity matrices H^1, H^2 between the true solutions $\mathbf{\Pi}^1, \mathbf{V}_x, \mathbf{\Pi}^2, \mathbf{V}_{yz}$ and the initial estimation output by matrix factorization $\hat{\mathbf{\Pi}}^1, \hat{\mathbf{V}}_x, \hat{\mathbf{\Pi}}^2, \hat{\mathbf{V}}_{yz}$:

$$\begin{aligned} \mathbf{L} &= \mathbf{\Pi}^1 \mathbf{V}_x = \hat{\mathbf{\Pi}}^1 H^1 (H^1)^{-1} \hat{\mathbf{V}}_x, \\ \mathbf{M} &= \mathbf{\Pi}^2 \mathbf{V}_{yz} = \hat{\mathbf{\Pi}}^2 H^2 (H^2)^{-1} \hat{\mathbf{V}}_{yz}. \end{aligned} \tag{42}$$

where $H^1 \in \mathbb{R}^{K \times K}$ and $H^2 \in \mathbb{R}^{2K \times 2K}$.

Now, the problem becomes to find H^1, H^2. Note that we have orthogonality constraints on each camera projection matrix R_n, which further impose constraints on $\mathit{\Pi}_n$. Thus, it can be used to partially estimate the ambiguity matrices H^1, H^2. Since the factorized matrix, *i.e.* $\mathbf{L}$ and $\mathbf{M}$, are the stacked 2D keypoints for all the images, thus H^1 and H^2 obtained from one image must satisfy the orthogonality constraints on other images, hence we use $\mathit{\Pi}_n \in \mathbb{R}^{2 \times 3K}$ (*i.e.* from image n) for our derivation.

Let $\hat{\mathit{\Pi}}_n = [\hat{\mathit{\Pi}}_n^1, \hat{\mathit{\Pi}}_n^2] = \begin{bmatrix} \hat{\pi}_n^{1,1:K}, \hat{\pi}_n^{1,K+1:3K} \\ \hat{\pi}_n^{2,1:K}, \hat{\pi}_n^{2,K+1:3K} \end{bmatrix}$, where $\hat{\pi}_n^{1,1:K}, \hat{\pi}_n^{2,1:K} \in \mathbb{R}^{1 \times K}$ are the first K columns of the first and second rows of $\hat{\mathit{\Pi}}_n$, and $\hat{\pi}_n^{1,K+1:3K}, \hat{\pi}_n^{2,K+1:3K} \in \mathbb{R}^{1 \times 2K}$ are the last $2K$ columns of the first and second rows of $\hat{\mathit{\Pi}}_n$, respectively. Thus, Eq. (42) implies:

$$\begin{aligned} L_n &= \hat{\mathit{\Pi}}_n^1 H^1 (H^1)^{-1} \hat{\mathbf{V}}_x = \begin{bmatrix} r_n^{11} \\ r_n^{21} \end{bmatrix} \mathbf{z}_n \mathbf{V}_x, \\ M_n &= \hat{\mathit{\Pi}}_n^2 H^2 (H^2)^{-1} \hat{\mathbf{V}}_{yz} = \begin{bmatrix} r_n^{1,2:3} \\ r_n^{2,2:3} \end{bmatrix} (\mathbf{z}_n \otimes I_2) \mathbf{V}_{yz}, \end{aligned} \tag{43}$$

where $L_n, M_n \in \mathbb{R}^{2 \times P}$ are the n'th double-row of $\mathbf{L}, \mathbf{M}$. $[r_n^{11}, r_n^{12}]^\top$ is the first column of the camera projection matrix of the n'th image R_n, and $[(r_n^{1,2:3})^\top, (r_n^{2,2:3})^\top]^\top$ is the second and third columns of R_n.

Let $h_k^1 \in \mathbb{R}^{K \times 1}, h_k^2 \in \mathbb{R}^{2K \times 2}$ be the kth column and double-column of H^1, H^2, respectively. Then, from Eq. (43), we get:

$$\begin{aligned} \hat{\mathit{\Pi}}_n^1 h_k^1 &= \begin{bmatrix} \hat{\pi}_n^{1,1K} \\ \hat{\pi}_n^{2,1K} \end{bmatrix} h_k^1 = z_{nk} \begin{bmatrix} r_n^{11} \\ r_n^{21} \end{bmatrix}, \\ \hat{\mathit{\Pi}}_n^2 h_k^2 &= \begin{bmatrix} \hat{\pi}_n^{1,K+1:3K} \\ \hat{\pi}_n^{2,K+1:3K} \end{bmatrix} h_k^2 = z_{nk} \begin{bmatrix} r_n^{1,2:3} \\ r_n^{2,2:3} \end{bmatrix}. \end{aligned} \tag{44}$$

By merging the equations of Eq. (44) together, R_n can be represented by:

$$[\hat{\mathit{\Pi}}_n^1 h_k^1, \hat{\mathit{\Pi}}_n^2 h_k^2] = z_{nk} R_n. \tag{45}$$

Remark 3 Similar to the rigid symmetry case in Eq. (29), Eq. (45) indicates that there is no rotation ambiguity in the symmetric direction (i.e., the x direction). The rotation ambiguities only exist in the yz-plane (*i.e.* the non-symmetric plane).

7.3 Solve the Ambiguities in Symmetry Prior Free Matrix Factorization Method

The main idea to solve the ambiguities here is similar to that in Sym-RSfM, *i.e.* we exploit the orthogonality constraints $R_n R_n^\top = I$. Using Eq. (45) and the orthogonality constraints, we have:

$$[\hat{\mathit{\Pi}}_n^1 h_k^1, \hat{\mathit{\Pi}}_n^2 h_k^2][\hat{\mathit{\Pi}}_n^1 h_k^1, \hat{\mathit{\Pi}}_n^2 h_k^2]^\top = z_{nk}^2 I \tag{46}$$

Remark 4 The main difference of the derivations from the orthogonality constraints between the rigid and non-rigid cases is that, for the rigid case, the dot product of each row of R_n is equal to 1, while for the non-rigid case, *i.e.* Eq. (46), the dot product on each row of $\mathit{\Pi}_n$ gives us a unknown value z_{nk}^2. But note that the unknown z_{nk}^2 is the same for the dot products on Row 1, and Row 2, of $\mathit{\Pi}_n$, this suggests that we can eliminate z_{nk}^2 by subtracting the dot product on the both rows of $\mathit{\Pi}_n$.

Next, we substitute Eq. (44) into Eq. (46), then eliminate the unknown z_{nk}^2 as suggested by Remark 4. After several derivations (which is detailed in the supplementary material), we arrive at similar $A_n \mathbf{x} = 0$ equations as Sym-RSfM in Eq. (47), where $\otimes$ denotes Kronecker product.

Stacking A_n for all the images yields the constraints:

$$\begin{aligned} &\mathbf{A}\mathbf{x} = 0, \\ &\mathbf{A} = [A_1^\top, ..., A_N^\top]^\top. \end{aligned} \tag{48}$$

$$A_n\mathbf{x}=0, \qquad \mathbf{x}=[\text{vec}(h_k^1 h_k^{1\top})^\top, \text{vec}(h_k^2 h_k^{2\top})^\top]^\top,$$
$$A_n=\begin{bmatrix}\hat{\pi}_n^{1,1:K}\otimes\hat{\pi}_n^{1,1:K}-\hat{\pi}_n^{2,1:K}\otimes\hat{\pi}_n^{2,1:K}, & \hat{\pi}_n^{1,K+1:3K}\otimes\hat{\pi}_n^{1,K+1:3K}-\hat{\pi}_n^{2,K+1:3K}\otimes\hat{\pi}_n^{1,K+1:3K}\\ \hat{\pi}_n^{1,1:K}\otimes\hat{\pi}_n^{2,1:K}, & \hat{\pi}_n^{1,K+1:3K}\otimes\hat{\pi}_n^{2,K+1:3K}\end{bmatrix}. \tag{47}$$

At first sight, it seems that Eq. (48) are not sufficient to solve for the ambiguity matrix H due to rank insufficiency [48], *i.e.* the solution of $\mathbf{x}$ lies in the null space of $\mathbf{A}$ of dimensionality $(2K^2-K)$ [48]. However, later research [4] proved that this rank insufficiency was merely a "gauge freedom" because all legitimate solutions lying in this subspace (despite under-constrained) gave the same solutions for the 3D structure. More technically, the ambiguity of H corresponds only to a linear combination of H's column-triplet and a rotation on H [2]. This observation was exploited by Dai *et al.* in [12,13], where they showed that, up to the ambiguities aforementioned, $h_k h_k^\top$ can be solved by the intersection of three subspaces as we will describe in the following.

Following the strategy in [12,13], we have the intersection of subspaces conditions shown in Eq. (49).

$$\left\{\mathbf{A}\begin{bmatrix}\text{vec}(h_k^1 h_k^{1\top})\\ \text{vec}(h_k^2 h_k^{2\top})\end{bmatrix}=0\right\}\cap\left\{\begin{matrix}h_k^1 h_k^{1\top}\succeq 0\\ h_k^2 h_k^{2\top}\succeq 0\end{matrix}\right\}\cap\left\{\begin{matrix}\text{rank}(h_k^1 h_k^{1\top})=1\\ \text{rank}(h_k^2 h_k^{2\top})=2\end{matrix}\right\}. \tag{49}$$

The first subspace comes from Eq. (48), *i.e.* the solutions of the Eq. (48) lie in the the null space of $\mathbf{A}$ of dimensionality $(2K^2-K)$ [48]. The second subspace requires that $h_k^1 h_k^{1\top}$ and $h_k^2 h_k^{2\top}$ are positive semi-definite. The third subspace comes from the fact that h_k^1 is of rank 1 and h_k^2 is of rank 2.

Note that as stated in [12,13], Eq. (49) imposes all the necessary constraints on $\mathbf{x}$. There is no difference in the recovered 3D structures using the different solutions that satisfy Eq. (49).

We can obtain a solution of $\mathbf{x}$, under the condition of Eq. (49), by standard semi-definite programming:

$$\begin{aligned}&\min ||h_k^1 h_k^{1\top}||_* + ||h_k^2 h_k^{2\top}||_*\\ \text{s. t.}\quad & h_k^1 h_k^{1\top}\succeq 0, \quad h_k^2 h_k^{2\top}\succeq 0,\\ & \mathbf{A}[\text{vec}(h_k^1 h_k^{1\top})^\top, \text{vec}(h_k^2 h_k^{2\top})^\top]^\top=0,\end{aligned} \tag{50}$$

where $||\cdot||_*$ indicates the trace norm.

7.4 Recovering the 3D Structure and the Camera Rotation Matrix

After solving h_k^1 and h_k^2 by Eq. (50), Eq. (45) (*i.e.* $[\hat{\Pi}_n^1 h_k^1,\ \hat{\Pi}_n^2 h_k^2]=z_{nk}R_n$) implies that the camera projection matrix R_n can be obtained by normalizing the two rows of $[\hat{\Pi}_n^1 h_k^1, \hat{\Pi}_n^2 h_k^2]$ to have a unit ℓ_2 norm [12,13]:

$$R_n=[\hat{\Pi}_n^1 h_k^1, \hat{\Pi}_n^2 h_k^2]/|||[\hat{\Pi}_n^1 h_k^1, \hat{\Pi}_n^2 h_k^2]||_2. \tag{51}$$

Then, $\mathbf{R}$ is constructed by $\mathbf{R}=\text{blkdiag}([R_1,...,R_N])$.

When the camera parameters are obtained, we can solve for the 3D structure by adopting the methods in [12,13], *i.e.* by minimizing a *low-rank* constraint on rearranged more compact $\mathbf{S}^\sharp$ under the orthographic model.

Similar to [12,13], the structure $\mathbf{S}$ can be estimated by:

$$\begin{aligned}&\min ||\mathbf{S}^\sharp||_*\\ \text{s. t.}\quad & [\mathbf{Y},\mathbf{Y}^\dagger]=\mathbf{R}[\mathbf{S},\mathcal{A}_N\mathbf{S}] \quad \mathbf{S}^\sharp=[\mathcal{P}_x,\mathcal{P}_y,\mathcal{P}_z](I_3\otimes\mathbf{S}),\end{aligned} \tag{52}$$

where $\mathcal{A}_N=I_N\otimes\mathcal{A}$, $\mathbf{S}=[S_1^\top,...,S_N^\top]^\top\in\mathbb{R}^{3N\times P}$. $\mathcal{P}_x,\mathcal{P}_y,\mathcal{P}_z\in\mathbb{R}^{N\times 3N}$ are the row-permutation matrices of 0 and 1 that select $(I_3\otimes\mathbf{S})$ to form $\mathbf{S}^\sharp$, *i.e.* $\mathcal{P}_x(i,3i-2)=1,\mathcal{P}_y(i,3i-1)=1,\mathcal{P}_z(i,3i)=1$ for $i=1,...,N$. Finally, $\mathbf{S}^\sharp\in\mathbb{R}^{N\times 3P}$ is rearranged more compact 3D structure, *i.e.*

$$\mathbf{S}^\sharp=\begin{bmatrix}x_{11},\ ...,\ x_{1P},\ y_{11},\ ...,\ y_{1P},\ z_{11},\ ...,\ z_{1P}\\ \vdots\qquad\quad \vdots\quad \vdots\qquad\quad \vdots\quad \vdots\qquad\quad \vdots\\ x_{N1},\ ...,\ x_{NP},\ y_{N1},\ ...,\ y_{NP},\ z_{N1},\ ...,\ z_{NP}\end{bmatrix}. \tag{53}$$

The algorithm for symmetry prior free matrix factorization (without occlusions) is summarized in Algorithm 4. Similar to the Sym-RSfM, Algorithm 4 can

Algorithm 4: Symmetry Prior Free Matrix Factorization Algorithm (Without Occlusions).

Input: The stacked keypoint sets $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ from N images (without occlusions).
Output: The 3D structure S_n and the camera matrix R_n for each image.

1 Change the coordinates to decouple the symmetry constraints by Eq. (40).
2 Get $\hat{\mathbf{\Pi}}^1,\hat{\mathbf{\Pi}}^2,\hat{\mathbf{V}}_x,\hat{\mathbf{V}}_{yz}$ by doing SVD on $\mathbf{L},\mathbf{M}$, *i.e.* Eq. (42).
3 Solve the ambiguities h_k^1 and h_k^2 by Eq. (50).
4 Get R_n by Eq. (51), then $\mathbf{R}=\text{blkdiag}([R_1,...,R_N])$.
5 Recover S_n by Eq. (52).

produce a good non-rigid SfM initialization for Algorithm 1 (i.e. Step 2 of Algorithm 1) if occluded keypoints exist. The full Sym-PriorFree method with occlusion reasoning is summarized in Fig. 4.

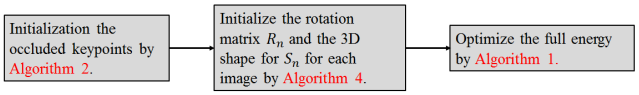

Fig. 4 The flowchart of the full non-rigid Sym-PriorFree method with occlusion reasoning.

8 The Symmetric EM-PPCA Method for Non-Rigid Structure from Motion

In this section, we discuss another approach to exploit symmetry in non-rigid SfM, which is quite different from the Sym-PriorFree method proposed in Sect. 7.

The method proposed in this section follows the other idea to address the additional ambiguities in non-rigid SfM, as discussed in Sect. 3. Specifically, we follow the idea of the Expectation-Maximization Probabilistic Principal Component Analysis (EM-PPCA) method to regularize the problem, *by imposing a Gaussian prior on the (deformation) coefficient to eliminate the additional gauge freedom* [44]. We name our algorithm as Symmetric EM-PPCA (Sym-EM-PPCA).

In addition, being different with Sym-PriorFree which is a direct matrix factorization method extending the rigid one (*i.e.* Sym-RSfM in Sect. 6), Sym-EM-PPCA is an indirect method which iteratively optimizes a specific energy function (or marginal probability) and takes the output of Sym-RSfM as initialization.

As an iterative method, Sym-EM-PPCA can naturally deal with the occluded keypoints using the same idea in Sect. 5, *i.e. by treating the occluded keypoints as latent variables and update them after optimizing all the unknown parameters*. In fact, we use a variant of Algorithm 1, namely Algorithm 1 fulfilled by Sym-RSfM (Algorithm 3), as the initialization when the occlusion exists.

In the rest of this section, we first discuss the problem formulation in Sect. 8.1. In Sect. 8.2, we give an EM algorithm to optimize the problem. Finally, we detail the initialization of the EM algorithm in Sect. 8.3. We assume occlusion applies in this section. The keypoint matrices are reshaped/vectorized (using blackboard bold characters, *e.g.* $\mathbb{Y}_n$) for convenient derivation. We use the same weak-perspective camera and explicitly model translation as [44].

8.1 Problem Formulation

In EM-PPCA [44], Bregler *et al.* assume that the 3D structure is represented by a mean structure $\bar{S}$ plus a non-rigid deformation. Suppose there are P keypoints on the structure, the non-rigid model of EM-PPCA is:

$$\mathbb{Y}_n = G_n(\bar{\mathbb{S}} + \mathbf{V}z_n) + \mathbb{T}_n + N_n, \tag{54}$$

where $\mathbb{Y}_n \in \mathbb{R}^{2P\times 1}, \bar{\mathbb{S}} \in \mathbb{R}^{3P\times 1}$, and $\mathbb{T}_n \in \mathbb{R}^{2P\times 1}$ are the stacked vectors of 2D keypoints, 3D mean structure and translations. $G_n = I_P \otimes c_n R_n$, in which c_n is the scale parameter for weak perspective projection, $\mathbf{V} = [\mathbb{V}_1, ..., \mathbb{V}_K] \in \mathbb{R}^{3P\times K}$ is the grouped K deformation bases, $z_n \in \mathbb{R}^{K\times 1}$ is the coefficient of the K bases, and N_n is the Gaussian noise $N_n \sim \mathcal{N}(0, \sigma^2 I)$.

Extending Eq. (54) to our symmetry problem in which there are P keypoint pairs $\mathbb{Y}_n$ and $\mathbb{Y}_n^\dagger$, we have:

$$\begin{aligned}\mathbb{Y}_n &= G_n(\bar{\mathbb{S}} + \mathbf{V}z_n) + \mathbb{T}_n + N_n,\\ \mathbb{Y}_n^\dagger &= G_n(\bar{\mathbb{S}}^\dagger + \mathbf{V}^\dagger z_n) + \mathbb{T}_n + N_n.\end{aligned} \tag{55}$$

As before, we assume that the object is symmetric along the x-axis which implies that the relationship between $\bar{\mathbb{S}}$ and $\bar{\mathbb{S}}^\dagger$, $\mathbf{V}$ and $\mathbf{V}^\dagger$ are:

$$\bar{\mathbb{S}}^\dagger = \mathcal{A}_P\bar{\mathbb{S}}, \qquad \mathbf{V}^\dagger = \mathcal{A}_P\mathbf{V}, \tag{56}$$

where $\mathcal{A}_P = I_P \otimes \mathcal{A}$ (recall that $\mathcal{A} = \mathrm{diag}([-1, 1, 1])$) and $I_P \in \mathbb{R}^{P\times P}$ is an identity matrix. Thus, we have[2]:

$$\begin{aligned}P(\mathbb{Y}_n|z_n, G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbb{T}) &= \mathcal{N}(G_n(\bar{\mathbb{S}} + \mathbf{V}z_n) + \mathbb{T}_n, \sigma^2 I),\\ P(\mathbb{Y}_n^\dagger|z_n, G_n, \bar{\mathbb{S}}, \mathbf{V}^\dagger, \mathbb{T}) &= \mathcal{N}(G_n(\mathcal{A}_P\bar{\mathbb{S}} + \mathbf{V}^\dagger z_n) + \mathbb{T}_n, \sigma^2 I).\end{aligned} \tag{57}$$

Following Bregler *et al.* [44], we introduce a prior $P(z_n)$ on the coefficient variable z_n. This prior is a zero mean unit variance Gaussian. It is used for (partly) regularizing the inference task but also for dealing with the ambiguities between basis coefficients z_n and bases $\mathbf{V}$, as mentioned above (when [44] was published it was not realized that these are "gauge freedoms"). This enables us to treat z_n as the hidden variable and use the EM algorithm to estimate the structure and camera viewpoint parameters. The formulation of the problem, in terms of Gaussian distributions (or, more technically, the use of conjugate priors) means that both steps of the EM algorithm are straightforward to implement.

[2] Note that we set hard constraints on $\bar{\mathbb{S}}$ and $\bar{\mathbb{S}}^\dagger$, *i.e.* replace $\bar{\mathbb{S}}^\dagger$ by $\mathcal{A}_P\bar{\mathbb{S}}$ in Eq. (57), because it can be guaranteed by our Sym-RSfM initialization in Sect. 6. While the initialization on $\mathbf{V}$ and $\mathbf{V}^\dagger$ by PCA cannot guarantee such a desirable property, thus a Language multiplier term is used for the constraint on $\mathbf{V}$ and $\mathbf{V}^\dagger$ in the following Eq. (61).

Remark 5 Our Sym-EM-PPCA method is a natural extension of the method in [44] to maximize the marginal probability $P(\mathbb{Y}_n, \mathbb{Y}_n^\dagger | G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T})$ with a Gaussian prior on z_n and a Language multiplier term (*i.e.* a regularization term) on $\mathbf{V}, \mathbf{V}^\dagger$. This can be solved by *general EM* algorithm [5], where both the **E** and **M** steps take simple forms because the underlying probability distributions are Gaussians (due to conjugate Gaussian prior).

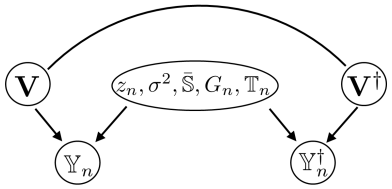

Fig. 5 The graphical model of the variables and parameters.

Algorithm 5: The Symmetry-EM-PPCA Algorithm (With Occlusions).

Input: The stacked keypoint sets from all the N images $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ with occluded points, in which each occluded point is set to $\mathbf{0}$ initially.
Output: The 3D structure S_n and the camera matrix R_n for each image, and the 2D keypoints with recovered occlusions.

Initialize R_n, $\bar{S} = S$, and the occluded keypoints in $\mathbf{Y}$ and $\mathbf{Y}^\dagger$ by Algorithms 1 and 3.
Initialize $c_n = 1$, and t_n by Eq. (63).
Initialize $\mathbf{V}$ and $\mathbf{V}^\dagger$ by Eq. (64).
Maximize $P(\mathbb{Y}_n, \mathbb{Y}_n^\dagger | G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T})$ by EM:
repeat
  E-Step: Update the statistics of z_n, *i.e.* μ_n and ϕ_n, by Eqs. (59) and (60).
  M-Step: Update the unknown parameters in θ by optimizing Eq. (61).
  Update the occluded keypoint by Eq. (62), and fill them in $\mathbb{Y}_n$ and $\mathbb{Y}_n^\dagger$.
until $P(\mathbb{Y}_n, \mathbb{Y}_n^\dagger | G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T})$ *converge*;

8.2 EM Algorithm for Optimization

In the following, we introduce an EM algorithm to maximize the marginal probability $P(\mathbb{Y}_n, \mathbb{Y}_n^\dagger | G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T})$.

E-Step: This step is to get the statistics of z_n from its posterior. Let the prior on z_n be $P(z_n) = \mathcal{N}(0, I)$ as in [44]. Then, we have $P(z_n)$, $P(\mathbb{Y}_n | z_n; \sigma^2, \bar{\mathbb{S}}, \mathbf{V}, G_n, \mathbb{T}_n)$ and $P(\mathbb{Y}_n^\dagger | z_n; \sigma^2, \bar{\mathbb{S}}, \mathbf{V}^\dagger, G_n, \mathbb{T}_n)$, which do not provide the complete posterior distribution directly. Fortunately, the conditional dependence of the variables shown in Fig. 5 (graphical model) implies that the posterior of z_n can be calculated by:

$$\begin{aligned}
&P(z_n | \mathbb{Y}_n, \mathbb{Y}_n^\dagger; \sigma^2, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, G_n, \mathbb{T}_n) \\
\sim &P(z_n, \mathbb{Y}_n, \mathbb{Y}_n^\dagger | \sigma^2, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, G_n, \mathbb{T}_n) \\
= &P(\mathbb{Y}_n | z_n; \sigma^2, \bar{\mathbb{S}}, \mathbf{V}, G_n, \mathbb{T}_n) P(\mathbb{Y}_n^\dagger | z_n; \sigma^2, \bar{\mathbb{S}}, \mathbf{V}^\dagger, G_n, \mathbb{T}_n) P(z_n) \\
\equiv &\mathcal{N}(z_n | \mu_n, \Sigma_n). \qquad (58)
\end{aligned}$$

The last equation of Eq. (58) is obtained by the fact that the prior and the conditional distributions of z_n are all Gaussians (conjugate prior), where we denote the mean and variance of z_n as μ_n and Σ_n, respectively, *i.e.* :

$$\begin{aligned}
\mu_n \equiv &E[z_n] = \gamma \mathbf{V}^\top G_n^\top (\mathbb{Y}_n - G_n \bar{\mathbb{S}} - \mathbb{T}_n) + \\
&\gamma \mathbf{V}^{\dagger\top} G_n^\top (\mathbb{Y}_n^\dagger - G_n \mathcal{A}_P \bar{\mathbb{S}} - \mathbb{T}_n), \qquad (59) \\
\Sigma_n \equiv &E[z_n z_n^\top] - E[z_n] E[z_n]^\top = \sigma^2 \gamma^{-1}. \qquad (60)
\end{aligned}$$

where $\gamma = (\mathbf{V}^\top G_n^\top G_n \mathbf{V} + \mathbf{V}^{\dagger\top} G_n^\top G_n \mathbf{V}^\dagger + \sigma^2 I)^{-1}$.

M-Step: This is to maximize the joint likelihood, which is similar to the coordinate descents in the Sym-RSfM and the Sym-PriorFree methods in the previous sections. The complete log-likelihood $\mathcal{Q}(\theta)$ is:

$$\begin{aligned}
&\mathcal{Q}(\theta) \\
= &\lambda ||\mathbf{V}^\dagger - \mathcal{A}_P \mathbf{V}||^2 - \sum_n \ln P(\mathbb{Y}_n, \mathbb{Y}_n^\dagger | z_n; G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T}_n, \sigma^2) \\
= &\lambda ||\mathbf{V}^\dagger - \mathcal{A}_P \mathbf{V}||^2 - \sum_n \ln P(\mathbb{Y}_n | z_n; G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbb{T}_n, \sigma^2) - \\
&\sum_n \ln P(\mathbb{Y}_n^\dagger | z_n; G_n, \bar{\mathbb{S}}, \mathbf{V}^\dagger, \mathbb{T}_n, \sigma^2), \\
&\text{s. t.} \qquad R_n R_n^\top = I, \qquad (61)
\end{aligned}$$

where $\theta = \{G_n, \bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger, \mathbb{T}_n, \sigma^2\}$.

The maximization of Eq. (61) is straightforward, *i.e.* taking the derivative of each unknown parameter in θ and equating it to 0. The update rule of each parameter is very similar to the original EM-PPCA [44] (except $\bar{\mathbb{S}}, \mathbf{V}, \mathbf{V}^\dagger$ should be updated jointly), which we put in the supplementary material.

After estimated all the other latent variables and the unknown parameters, we can further update the occluded keypoints $\{Y_{n,p}, (n,p) \in IVS\}, \{Y_{n,p}^\dagger, (n,p) \in IVS^\dagger\}$ by:

$$\begin{aligned}
Y_{n,p} &= R_n (\bar{S}_p + \mathbf{V}_p z_n) + t_n, \\
Y_{n,p}^\dagger &= R_n (\mathcal{A} \bar{S}_p + \mathbf{V}_p^\dagger z_n) + t_n, \qquad (62)
\end{aligned}$$

where $\bar{S}_p \in \mathbb{R}^{3\times 1}$ is the $[p, p+P, p+2P]$-th elements of $\mathbb{S}$, $\mathbf{V}_p \in \mathbb{R}^{3\times K}$ is the $[p, p+P, p+2P]$-th rows of $\mathbf{V}$, $t_n \in \mathbb{R}^{2\times 1}$ is the translation (which is the same for all the keypoints within the same image).

8.3 Initialization

The camera rotation matrix R_n, the mean shape $\bar{S}$, and the occluded points $Y_{n,p}, Y_{n,p}^\dagger$) can be initialized

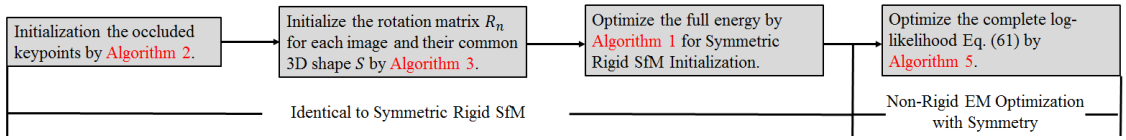

Fig. 6 The flowchart of the full non-rigid Sym-EM-PPCA method with occlusion reasoning.

by Sym-RSfM with coordinate descent, *i.e.* Algorithms 1 and 3. The scale c_n is initialized as 1, the translation t_n is initialized by:

$$t_n = \sum_p (Y_{n,p} - R_n \bar{S}_p + Y^{\dagger}_{n,p} - R_n \mathcal{A} \bar{S}_p). \tag{63}$$

Then, the deformation bases $\mathbf{V}$ and $\mathbf{V}^{\dagger}$ can be initialized by doing PCA on the residual of the 2D keypoints minus their rigid projections iteratively, *i.e.* :

$$\text{PCA on } (\mathbf{Y} - \mathbf{R}\bar{S} - \mathbf{1}_P^{\top} \otimes t_n),\ (\mathbf{Y}^{\dagger} - \mathbf{R}\bar{S}^{\dagger} - \mathbf{1}_P^{\top} \otimes t_n). \tag{64}$$

The algorithm for Sym-EM-PPCA is summarized in Algorithm 5, which makes use of the results from Sym-RSfM as initialization. We also summarize a algorithmic flowchart for the Sym-EM-PPCA method in Fig. 6.

9 Experimental Results

This paper discusses 3 different scenarios to reconstruct the 3D structure: (i) reconstruction from a single image using symmetry and the Manhattan assumptions (Sect. 4), (ii) reconstruction from multiple images using symmetric rigid SfM (Sect. 6), (iii) reconstruction from multiple images using symmetric non-rigid SfM (Sects. 7 and 8). The experiments are performed on the Pascal3D+ dataset. This contains object categories such as *aeroplane* and *car*. In Pascal3D+, the object categories are further divided into subtypes, such as *sedan* car, we show all the subtypes of the *car* category in Fig. 7[3]. For each object subtype, we estimate the 3D structure and the viewpoint. The 3D structure is specified by the 3D keypoints in Pascal3D+ [47] and we also have corresponding keypoints in the 2D images from Berkeley [6]. These are the same experimental settings used in [26].

For evaluation, we calculate the rotation error e_R and shape error e_S, as in [3, 12, 13, 18]. Note that Pascal3D+ provided only one common shape for each *subtype*, therefore, we report the shape error according to

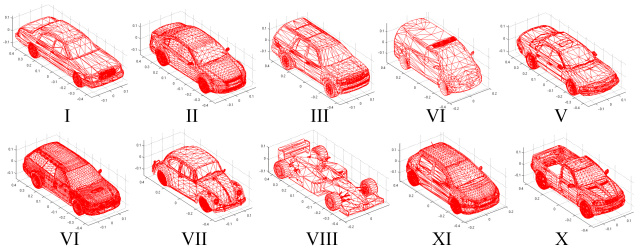

Fig. 7 Illustration of the 10 subtypes (denoted by the Roman numerals) for the *car* category in the Pascal3D+ dataset.

each *subtype*[4]. The 3D groundtruth and our 3D estimates may have different scales, so we normalize them before evaluation. For each shape S_n we use its standard deviations in X, Y, Z coordinates $\sigma_n^x, \sigma_n^y, \sigma_n^z$ for the normalization: $S_n^{\text{norm}} = 3S_n/(\sigma_n^x + \sigma_n^y + \sigma_n^z)$. To deal with the rotation ambiguity between the 3D groundtruth and our reconstruction, we use the Procrustes method [40] to align them. Assume we have $2P$ keypoints, *i.e.* P point pairs, the the rotation error e_R and shape error e_S can be calculated as:

$$e_R = \frac{1}{N} \sum_{n=1}^{N} ||R_n^{\text{aligned}} - R_n^*||_{\text{F}}, \tag{65}$$

$$e_S = \frac{1}{2NP} \sum_{n=1}^{N} \sum_{p=1}^{2P} ||S_{n,p}^{\text{norm aligned}} - S_{n,p}^{\text{norm}*}||_{\text{F}}, \tag{66}$$

where R_n^{aligned} and R_n^* are the recovered and the groundtruth camera projection matrix for image n. $S_{n,p}^{\text{norm aligned}}$ and $S_{n,p}^{\text{norm}*}$ are the normalized estimated and the normalized groundtruth structure for the p'th point of image n. R_n^{aligned} and R_n^*, $S_{n,p}^{\text{norm aligned}}$ and $S_{n,p}^{\text{norm}*}$ are aligned by the Procrustes method [40].

In the following, we perform the experiments on single image in Sect. 9.1, multiple images with rigid deformations in Sect. 9.2, and multiple images with non-rigid deformations in Sect. 9.3. The experiments demonstrate that, for single image case, our results are very promising if all keypoints are visible without noisy annotations; for the multiple images reconstruction, our methods outperform the corresponding state-of-the-art

[3] For the subtypes of more categories, please refer to the Pascal3D+ official website at http://cvgl.stanford.edu/projects/pascal3d.html.

[4] For the rigid case, as we use the images from the same *subtype* as input (so that we can reasonably assume rigid deformation among them), therefore, we also report the rotation error according to *subtype* for the rigid experiments.

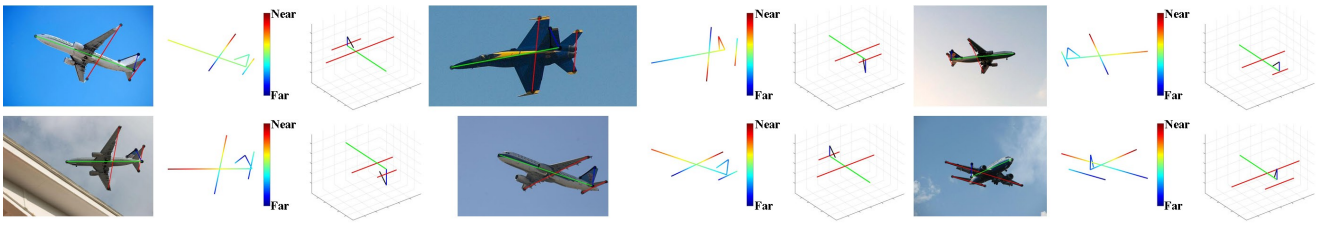

Fig. 8 Illustration of the reconstruction results for *aeroplane* using symmetry and Manhattan on *single image*. For each subfigure triplet, the first subfigure is the 2D image, the second and third subfigures are the 3D structure from the original and rectified viewpoints. The gauge freedoms of sign ambiguities can also be observed by comparing the rectified 3D reconstructions, *i.e.* the third subfigures. The Red, Green, Blue lines represent the 3 Manhattan directions (*i.e.* left-right, front-end, top-bottom), and other directions are represented by the Black lines (best view in color).

methods in most cases in both rigid and non-rigid settings, respectively.

9.1 Experiments on Single Image

We use *aeroplane*s for this experiment because the 3 Manhattan directions (*e.g.* left wing → right wing, nose → tail, and top rudder → bottom rudder) can be obtained directly, see Fig. 1. Also, *aeroplane*s are generally far away from the camera, implying that orthographic projection is a good approximation.

We selected 42 images with clear 3 Manhattan directions and with no occluded keypoints from the *aeroplane* category of the Pascal3D+ dataset, and evaluated the results by the Rotation Error and Shape Error (Eq. (66)). The shape error is obtained by comparing the reconstructed structure with their subtype groundtruth model of Pascal3D+ [47].

The *average rotation and shape errors* for *aeroplane* using symmetry and Manhattan on single image are *0.3210* and *0.6047*, respectively[5]. These results show that symmetry and Manhattan can give good results for single image reconstruction. Indeed, the performance is better than some of the structure from motion (SfM) methods which use multiple images, see Tables 1 and 2. But this is not a fair comparison, because these 42 images are selected to ensure that all the Manhattan axes are observed, while the SfM methods have been evaluated on all the aeroplane images. We also illustrate some reconstruction results in Fig. 8.

[5] As there is no baseline method for comparison, we also calculate the average rotation errors measured by averaged geodesic distance $\frac{1}{N}\sum_{n=1}^{N}||\log(R_n^{\text{aligned}\top}R_n^*)||_{\text{F}}/\sqrt{2}$, which represents the angle difference between two rotation matrices. The results show that the rotation error is *4.1766* degree in average.

9.2 Experiments on Multiple Images with Rigid Deformation

In this section, we estimate the 3D structures of each subtype and the orientations of all the images within that subtype for *aeroplane, bus, car, sofa, train, tv* in Pascal3D+ [47]. Note that Pascal3D+ provides a single 3D shape for each object subtype rather than for each object. For example, it provides 10 subtypes such as *sedan, truck* for the *car* category, ignoring the shape variation within each subtype. Thus, we divide the images of the same category into subtypes, and then input the images of each subtype to our Sym-RSfM for the experiments. The outputs for this problem is the *common 3D structure for all the input images* (*i.e.* for the input subtype), and the *different rotation matrix for each image*. In order to be consistent with our input, we summarize both the shape and the rotation errors according to each *subtype*.

Following [26], images with more than 5 visible keypoints are used. Also, as in [26], we augment the images by left-right flipping for all the methods. The rotation error and the shape error are calculated by Eq. (66). The rigid SfM (RSfM) [42] and the more recent CSF method [18], which does not exploit symmetry, is used for comparison. Note that the CSF method [18] utilized smooth time-trajectories as initialization, which does not always hold in our application, because the input images of our application are not from a continuous video. Therefore, we also investigate the results from CSF method with random initialization. We report the CSF results with smooth prior as *CSF (S)* and the *best* results with 10 random initialization as *CSF (R)*.

The results (mean rotation and shape errors) on *aeroplane, bus, car, sofa, train, tv* are shown in Tables 1 and 2, where the subtypes are indexed by Roman numerals.

Tables 1 and 2 show that our method (Sym-RSfM) outperforms the baseline methods for almost all cases. The cases that our method does not perform as the

Table 1 Rigid Deformation. The mean *rotation* errors for *aeroplane, bus, car, sofa, train, tv*, calculated using the images from the same subtype (denoted by the Roman numerals) as input. CSF (S) means the CSF [18] results with smooth time-trajectories initialization, CSF (R) means the *best* results of CSF [18] with random initialization in 10 runs.

	aeroplane							**bus**					
	I	II	III	IV	V	VI	VII	I	II	III	IV	V	VI
RSfM	0.56	1.19	0.76	0.82	0.96	1.22	0.81	0.68	0.73	0.56	0.48	0.63	0.93
CSF (S)	0.95	**0.67**	0.93	0.87	0.87	0.89	0.88	0.78	0.86	0.90	0.91	0.97	0.98
CSF (R)	0.94	0.79	0.93	0.91	0.95	0.98	0.97	0.77	0.89	0.89	0.92	1.08	1.03
Sym-RSfM	**0.25**	1.09	**0.47**	**0.48**	**0.37**	**0.80**	**0.56**	**0.38**	**0.22**	**0.21**	**0.23**	**0.41**	**0.78**

	car										**sofa**	
	I	II	III	IV	V	VI	VII	VIII	IX	X	I	II
RSfM	0.94	0.79	0.72	0.69	0.81	0.89	0.63	1.08	0.68	0.70	1.03	0.89
CSF (S)	1.00	0.85	0.96	0.92	0.90	1.08	0.85	**0.99**	0.97	1.01	1.01	0.73
CSF (R)	0.93	0.87	0.89	0.83	0.91	0.76	0.86	1.02	0.88	0.74	0.89	0.77
Sym-RSfM	**0.43**	**0.35**	**0.32**	**0.42**	**0.29**	**0.40**	**0.35**	1.04	**0.27**	**0.37**	**0.73**	**0.12**

	sofa				**train**				**tv**			
	III	IV	V	VI	I	II	III	IV	I	II	III	IV
RSfM	0.89	1.36	1.44	1.45	1.69	0.90	1.39	1.11	**0.64**	0.56	0.43	1.43
CSF (S)	0.90	1.19	1.27	1.36	0.88	1.27	**0.84**	**0.87**	0.90	0.80	0.87	**0.77**
CSF (R)	0.97	1.09	**1.15**	1.26	0.91	**0.77**	0.94	0.90	0.93	0.82	0.86	0.92
Sym-RSfM	**0.15**	**0.83**	**1.15**	**1.14**	**0.48**	1.53	1.36	1.38	1.10	**0.20**	**0.24**	0.86

Table 2 Rigid Deformation. The mean *shape* errors for *aeroplane, bus, car, sofa, train, tv*, calculated using the images from the same subtype (denoted by the Roman numerals) as input.

	aeroplane							**bus**					
	I	II	III	IV	V	VI	VII	I	II	III	IV	V	VI
RSfM [42]	0.44	1.17	0.52	**0.29**	0.76	0.54	0.61	1.29	1.27	1.16	0.97	1.52	1.21
CSF (S)	1.25	**0.36**	1.42	0.84	0.33	0.47	**0.59**	1.11	**0.39**	0.56	**0.16**	3.02	**0.44**
CSF (R)	0.25	0.44	0.34	1.40	0.58	1.73	0.69	0.99	1.06	1.33	0.88	1.94	2.00
Sym-RSfM	**0.19**	0.88	**0.27**	0.34	**0.33**	**0.30**	0.62	**0.68**	0.58	**0.35**	0.24	**0.76**	0.47

	car										**sofa**	
	I	II	III	IV	V	VI	VII	VIII	IX	X	I	II
RSfM [42]	1.48	1.49	1.33	1.38	1.45	1.39	1.21	1.81	1.22	1.07	2.50	1.09
CSF (S)	1.06	2.33	1.15	1.17	1.36	1.17	1.03	1.10	2.03	0.99	1.78	0.24
CSF (R)	1.34	1.07	1.03	1.16	1.18	1.26	0.88	**0.90**	1.65	1.13	**0.76**	0.25
Sym-RSfM	**1.03**	**0.96**	**0.95**	**1.07**	**0.89**	**1.00**	**0.81**	1.66	**0.88**	**0.71**	2.27	**0.22**

	sofa				**train**				**tv**			
	III	IV	V	VI	I	II	III	IV	I	II	III	IV
RSfM [42]	1.49	1.60	3.44	2.56	1.68	0.39	0.28	0.22	0.23	0.88	0.64	1.77
CSF (S)	3.14	1.54	2.74	1.55	0.83	0.85	0.25	0.26	0.66	0.77	0.34	0.34
CSF (R)	1.82	1.19	1.42	1.20	1.05	**0.37**	0.24	**0.17**	0.22	0.97	0.55	0.36
Sym-RSfM	**0.40**	**1.07**	**0.87**	**1.14**	**0.73**	0.61	**0.13**	0.24	**0.09**	**0.29**	**0.32**	**0.14**

best may be because Pascal3D+ assumes the shapes from objects within the same subtype are very similar to each other, but this might be violated sometimes. In summary, the good performance of our method is due to exploiting the symmetry constraints, which is an intrinsic property of most man-made objects.

9.3 Experiments on Multiple Images with Non-Rigid Deformation

In this section, we construct the 3D keypoints for each image using the non-rigid model. Firstly, we follow the experimental setting in [26] and collect all images with more than 5 visible keypoints, we also augment the images by left-right flipping. Then, we do 10 iterations with rank 3 recovery to initialize the occluded/missing data. In this experiments, we use 3 deformation bases and set λ in Sym-EM-PPCA, *i.e.* in Eq. (61), as 1.

We tested our algorithm on Pascal *aeroplane, bus, car, sofa, train, tv* based on the mean shape and rotation errors as in Eq. (66). Unlike the rigid case, we input all the images from the same *category* (*e.g. car*), and the algorithm outputs *different 3D structure and rotation matrix for each image*. In the non-rigid case, we evaluate the rotation errors according to each *image* and summarize them by *category*. But for the shape results, as we lack the ground truth 3D annotations for each image in Pascal3D+ (which only provides one 3D model for each subtype), therefore we evaluate the shape errors according to each *subtype*.

The results are reported in Table 3. The results show that our method outperforms the baselines in general. But we note that our method is not as good as the baselines in some cases, especially for *tv*. The reasons might be: (i) the orthographic/weak-perspective projection is less accurate when the object is close to the camera.

Table 3 Non-Rigid Deformation. The mean *shape* and *rotation* errors for *aeroplane, bus, car, sofa, train, tv*. The Roman numerals indicates the index of subtypes for the mean shape error, and mRE is short for the mean rotation error. EP, PF, Sym-EP, Sym-PF are short for EM-PPCA [44], PriorFree [12,13], Sym-EM-PPCA, Sym-PriorFree, respectively.

	aeroplane							
	I	II	III	IV	V	VI	VII	mRE
EP	0.36	0.59	0.50	0.49	0.57	0.57	**0.45**	0.34
PF	0.99	1.08	1.13	1.15	1.22	1.10	1.11	0.52
Sym-EP	**0.33**	**0.53**	**0.46**	**0.43**	**0.51**	**0.53**	0.46	**0.31**
Sym-PF	0.57	0.76	0.84	0.76	0.73	0.61	0.79	0.46

	bus						
	I	II	III	IV	V	VI	mRE
EP	0.52	0.47	0.49	0.43	0.80	0.59	0.32
PF	0.62	1.56	1.75	1.59	2.09	1.70	0.47
Sym-EP	**0.28**	**0.25**	**0.33**	**0.33**	**0.65**	**0.46**	**0.21**
Sym-PF	0.69	0.68	0.74	0.74	0.99	0.82	0.35

	car										
	I	II	III	IV	V	VI	VII	VIII	IX	X	mRE
EP	1.10	1.01	1.09	1.05	1.03	1.07	0.99	1.46	1.00	0.85	0.39
PF	1.76	1.67	1.76	1.77	1.65	1.79	1.67	1.57	1.70	1.42	0.86
Sym-EP	**0.99**	**0.89**	**1.05**	**1.02**	**0.92**	**1.00**	**0.89**	**1.39**	**0.95**	**0.68**	**0.34**
Sym-PF	1.74	1.41	1.70	1.48	1.69	1.58	1.43	1.69	1.52	1.30	0.79

	sofa		
	I	II	III
EP	2.00	1.87	2.01
PF	1.71	1.41	1.46
Sym-EP	1.06	**0.70**	**1.00**
Sym-PF	**1.04**	0.74	1.04

	sofa			
	IV	V	VI	mRE
EP	1.98	2.36	1.81	0.78
PF	2.02	2.66	1.64	1.36
Sym-EP	**0.93**	1.61	**0.73**	**0.33**
Sym-PF	0.96	**1.13**	1.58	0.89

	train							
	I	II	III	IV	mRE	I	II	III
EP	**0.92**	0.60	0.48	0.47	0.97	0.51	**0.51**	**0.42**
PF	1.97	**0.52**	0.49	0.45	1.02	0.57	1.01	0.97
Sym-EP	0.97	0.57	0.49	**0.37**	**0.77**	**0.38**	0.56	0.51
Sym-PF	1.43	0.58	**0.44**	0.45	1.02	0.60	1.08	1.12

	tv		
	IV	mRE	
EP	0.30	**0.42**	
PF	0.66	0.80	
Sym-EP	0.49	0.52	
Sym-PF	**0.20**	0.75	

Although all the methods used the same suboptimal orthographic projection for these cases, it may deteriorate more on our model sometimes, since we model more constraints. (ii) It might be because that the 3D groundtruth in Pascal3D+, which neglects the shape variations in the same subtype, may not be accurate enough (*e.g.* it has only one 3D model for all the *sedan* cars).

In addition, the results of the EM-PPCA based methods (Sym-EM-PPCA and EM-PPCA) are generally better than those of the PriorFree methods (PriorFree and Sym-PriorFree). Apart from the imperfect 3D annotations in Pascal3D+, we hypothesize that this is due to that (i) the Gaussian prior assumption in the EM-PPCA based methods can possibly better represent the input data. (ii) We only have a limited number of the input keypoints, which leads insufficiency in the number of the maximally allowed deformation bases[6]. (iii) The post processing (*i.e.* , projection energy minimization) is important. The EM-PPCA methods perform EM algorithm to jointly optimize more parameters, which generalize to MLE used the PriorFree methods, and therefore, producing better results.

The performance for the non-rigid SfM in Table 3 is sometimes lower than that for the rigid SfM in Tables 1 and 2. But note that they are not directly comparable due to the difference in the input data, *i.e.* the non-rigid SfM uses all the images from the same *category* (*e.g. car*) as input, while rigid SfM only inputs the images within the same *subtype* (*e.g. sedan car*). In fact, the non-rigid SfM is a more difficult problem than the rigid SfM, which assumes non-rigid deformation between the input images and does not require additional subtype labels.

[6] As analyzed in Remark 10 and Eq. (38), the relationship between the number of allowed deformation bases K and the number of keypoint pairs P follows: $K \leq P/3$.

10 Towards Practical 3D Object Structure Reconstruction and Viewpoint Estimation

In this section, we discuss how to build a practical 3D object structure reconstruction and viewpoint estimation system in a fully automatic pipeline from an input image. In our previous experiments, we used the ground truth 2D keypoints as input to our algorithms, which are (mostly) perfectly symmetric. However, the imperfect annotations will inevitably arise when we use features detectors to localize the keypoints. Therefore, the experiments carried out in this section also demonstrate the robustness of our algorithms, especially when the input 2D annotations are not perfectly symmetric.

Firstly, we did a study to investigate what happens if the keypoints are not perfectly annotated. We simulated this by adding different amounts of Gaussian noise to the 2D ground truth annotations and then re-did the experiments. This was to ensure that our methods are robust to the imperfectly symmetric annotations. The results are shown in the supplementary material, which demonstrates that the performances of all the methods decrease overall as the amount of Gaussian noise increases. Nevertheless, our methods still outperform our baselines despite the noisy annotations. In other words, these results suggest that our methods are robust to imperfect annotations and hence suitable for practical use.

But to go further, we implement a system where the keypoints are extracted by features detectors, *e.g.* deep nets [9], to yield a practical and automatic system, as

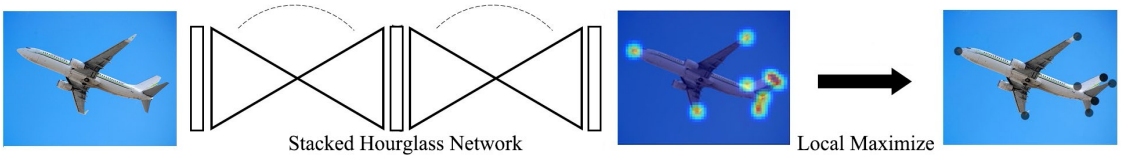

Fig. 6 Keypoint localization by stacked hourglass network, where two hourglass network units are used (best view in color).

shown in Fig. 5. We only lack one important step, *i.e.* how to localize *keypoints with semantic meanings* given an image as input. Note that the semantic meanings associated with the keypoints are necessary, as they are required to locate the symmetric pairs (*e.g.* left wheel - right wheel).

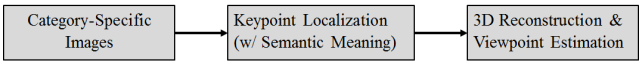

Fig. 5 The flowchart of the 3D structure reconstruction and viewpoint estimation in practice.

Fortunately, the state-of-the-art stacked hourglass network [36] satisfies our requirements. The stacked hourglass network was originally introduced for human pose estimation in order to localize the keypoints of human bodies, such as neck, left elbow, left wrist, *etc.* Each hourglass network unit is used to capture both local and global cues for keypoint localization. It is designed to have a symmetric bottom-up (encoder) and a top-down (decoder) inference at different scales (representing local and global information) and produces multiple heatmaps as output. These heatmaps can be resized to the same resolution of the input image, each of which represents the probabilities of the location for a specific keypoint over the whole image. The location of each keypoint can then be obtained as the global maxima of the corresponding heatmap, this also enables to determine the semantic meaning of each keypoint. Finally, [36] stacks multiple hourglass units to repeatedly refine the features for more precise keypoints localization.

It has been shown that the stacked hourglass network also works well for object keypoint localization [38]. Similar to [38], our stacked hourglass network includes two stacked hourglass network units, trained from scratch by the 2D annotations from Pascal3D+ dataset [47] with augmented annotations on ImageNet images. The procedure for localizing keypoints using the stacked hourglass network is shown in Fig. 6.

In the following, we conduct the experiments using the keypoints localized by the stacked hourglass network to enable practical 3D object structure reconstruction and viewpoint estimation. Note that the stacked hourglass network can automatically detect and localize the *self-occluded* keypoints, but it does not work well on the objects which are *occluded by other objects* or *truncated objects*, where parts are outside the image[7]. Therefore, similar to [38], we filtered out the *occluded-by-others* and *truncated* objects. Note that there is only one image for *sofa* subtype VI after filtering out undesirable objects, therefore we only conduct experiments on *sofa* subtype I - V. In addition, the Pascal3D+ dataset may annotate inconsistently on different object subtypes from the same category (*e.g.* there are 8 keypoints for CRT tvmonitors, but only 4 for LCD ones), we use the common keypoints across different subtypes for our experiments. This makes it infeasible to conduct the experiments on *tvmonitor*, as the common 4 keypoints are co-planar, leading mathematical degeneracy/failure for multiple algorithms. As a result, we report the performance on *aeroplane, bus, car, train* and *sofa* Subtype I - V in the following subsections.

10.1 Experiments on Multiple Images with Rigid Deformation

We conduct the experiments on the rigid SfM problem with the same experimental setup as that in Sect. 9.2, except that the 2D keypoints are detected by the stacked hourglass network and we do not use left-right flipping augmentation here.

Tables 4 and 5 show the results (mean rotation and shape errors) on *aeroplane, bus, car, sofa, train*, which demonstrate that our Sym-RSfM method outperforms the state-of-the-art methods in general for both mean rotation and shape errors. Interestingly, the results in Tables 4 and 5 for some subtypes are even better than those in Tables 1 and 2 which used manually labeled 2D annotations (though they are not directly compa-

[7] This is because the self-occluded information/features can be recovered by the training images from a different viewpoint, but the training data cannot exhaustively retain various occlusions introduced by other objects or various truncated types.

Table 4 Rigid Deformations with Hourglass Input. The mean *rotation* errors for *aeroplane, bus, car, sofa, train*, calculated using the images from the same subtype (denoted by the Roman numerals) as input. CSF (S) means the CSF [18] results with smooth time-trajectories initialization, CSF (R) means the *best* results of CSF [18] with random initialization in 10 runs.

	aeroplane							**bus**			
	I	II	III	IV	V	VI	VII	I	II	III	IV
RSfM	0.25	1.13	0.82	0.48	0.34	0.40	0.24	0.52	0.34	0.42	0.41
CSF (S)	0.93	**0.67**	0.89	1.01	0.89	0.92	0.82	0.32	0.17	0.16	0.12
CSF (R)	0.94	1.04	0.93	0.91	0.92	0.90	0.86	0.88	0.92	0.92	0.94
Sym-RSfM	**0.16**	1.03	**0.62**	**0.35**	**0.27**	**0.32**	**0.20**	**0.82**	**0.88**	**0.87**	**0.92**

	bus		**car**								
	V	VI	I	II	III	IV	V	VI	VII	VIII	IX
RSfM	0.53	0.37	0.39	0.25	0.29	0.23	0.27	0.36	0.30	1.30	0.28
CSF (S)	0.81	0.74	0.84	0.97	0.70	0.93	0.91	0.78	0.86	0.79	0.87
CSF (R)	0.73	0.69	0.93	0.87	0.87	0.85	0.87	0.76	0.83	1.03	0.88
Sym-RSfM	**0.28**	**0.17**	**0.19**	**0.09**	**0.09**	**0.13**	**0.09**	**0.10**	**0.16**	**0.27**	**0.09**

	car	**sofa**					**train**			
	X	I	II	III	IV	V	I	II	III	IV
RSfM	0.40	0.46	0.22	0.58	0.34	**0.25**	0.67	0.44	0.40	0.77
CSF (S)	0.88	0.95	0.79	1.26	0.42	0.42	0.91	0.72	0.86	0.94
CSF (R)	0.82	0.90	0.79	0.76	0.58	0.46	0.93	0.73	0.89	0.93
Sym-RSfM	**0.18**	**0.46**	**0.15**	**0.37**	**0.13**	0.38	**0.39**	**0.32**	**0.21**	**0.63**

Table 5 Rigid Deformations with Hourglass Input. The mean *shape* errors for *aeroplane, bus, car, sofa, train*, calculated using the images from the same subtype (denoted by the Roman numerals) as input.

	aeroplane							**bus**			
	I	II	III	IV	V	VI	VII	I	II	III	IV
RSfM	0.27	1.04	1.24	0.62	0.24	0.41	0.31	1.04	0.81	0.91	0.98
CSF (S)	0.47	**0.62**	0.52	2.00	0.29	0.22	0.37	1.64	0.80	1.30	1.34
CSF (R)	0.18	0.71	**0.39**	0.34	0.34	0.21	0.35	1.35	1.13	0.92	1.01
Sym-RSfM	**0.09**	0.72	0.58	**0.34**	**0.24**	**0.10**	**0.24**	**0.59**	**0.36**	**0.38**	**0.25**

	bus		**car**								
	V	VI	I	II	III	IV	V	VI	VII	VIII	IX
RSfM	1.16	1.21	0.91	0.58	0.75	0.61	0.67	0.83	0.61	1.41	0.63
CSF (S)	1.22	1.02	0.89	0.29	0.51	0.67	0.44	0.79	0.36	0.81	0.19
CSF (R)	1.12	1.36	0.96	0.35	0.77	0.61	0.64	0.69	0.60	0.61	0.52
Sym-RSfM	**0.52**	**0.40**	**0.23**	**0.10**	**0.14**	**0.07**	**0.14**	**0.19**	**0.14**	**0.27**	**0.07**

	car	**sofa**					**train**			
	X	I	II	III	IV	V	I	II	III	IV
RSfM	0.78	0.59	**0.28**	0.73	0.58	0.60	0.90	1.43	0.74	1.24
CSF (S)	0.62	2.48	0.34	0.85	0.36	0.52	1.07	1.22	**0.12**	0.86
CSF (R)	0.70	0.51	0.36	0.63	0.53	0.48	0.90	1.76	0.96	1.37
Sym-RSfM	**0.09**	**0.38**	0.29	**0.36**	**0.32**	**0.29**	**0.18**	**1.07**	0.27	**0.34**

rable)[8]. The reason may be that the keypoint localization network can automatically detect the self-occluded keypoints, therefore Tables 4 and 5 contain much less missing keypoints.

Figure 7 illustrates some reconstruction results from the Sym-RSfM method. It demonstrates that our method gives good reconstruction once the keypoints are well localized.

We also observe that, in some cases, the reconstructions do not perfectly match the 2D annotations. This is because, for rigid SfM, we estimate only one common 3D structure for all the input images (by assuming that the images from the same subtype are only up to some "fake" rigid deformation ambiguities). This imprecise assumption leads to imperfect matches of 3D structure and 2D annotations. This problem can be alleviated in non-rigid SfM, where we estimate one 3D structure for each object. We also summarize other failure cases for the rigid SfM. Conversely, the rigid SfM algorithms will not work well if the input keypoints are not well localized, and if all the input keypoints are co-planar. In addition, our method does not work well when the object is very near to the camera. This is because that the orthographic or weak-perspective camera assumption will be violated in this case. We leave the algorithm for the more precise (full) perspective camera as our future work.

[8] They are not directly comparable because (i) Tables 1 and 2 use 2D annotations from [6] (the same as those used in [26]), while the keypoint localization network for Tables 4 and 5 is trained on 2D annotations from Pascal3D+ [47]. (ii) We exclude the occluded-by-others and truncated objects in Tables 4 and 5 (the same as those in [38]) because the stacked hourglass network [36] does not produce satisfied results on those images.

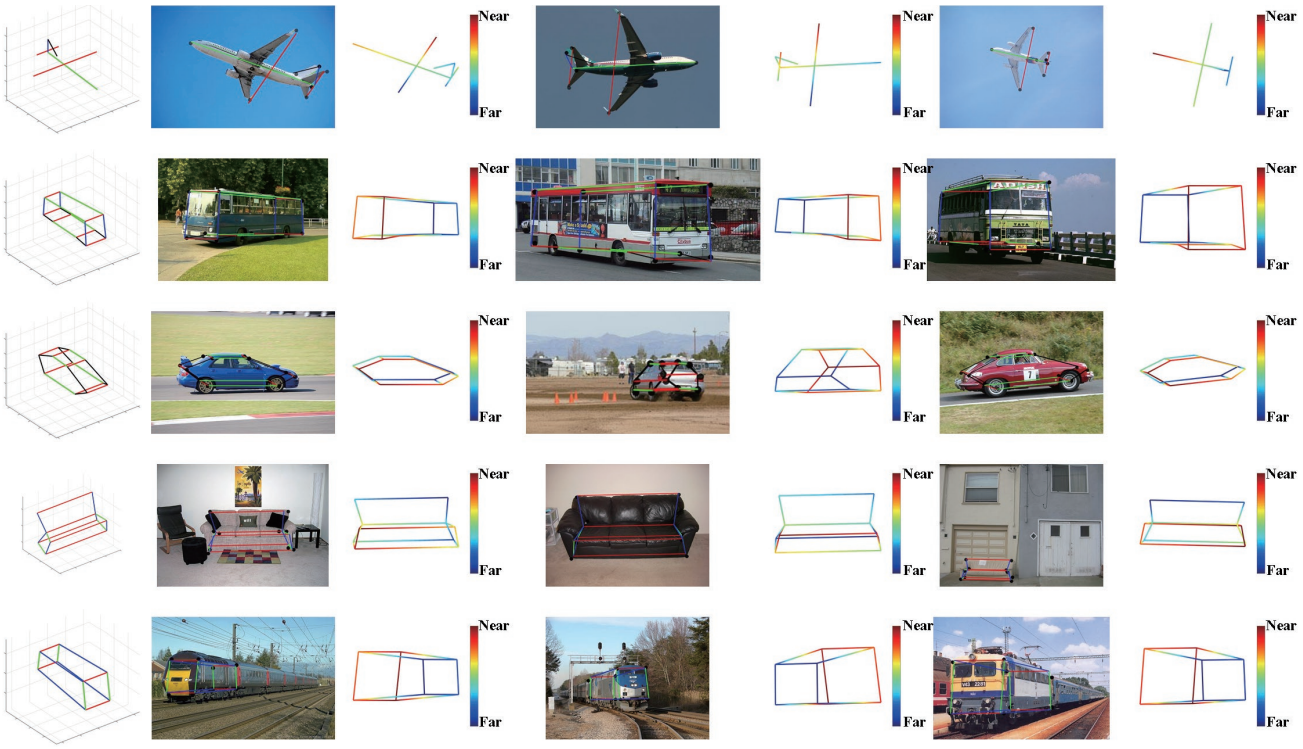

Fig. 7 The illustrations for 3D shapes and viewpoints estimated by Sym-RSfM for *aeroplane, bus, car, sofa,* and *train*. The first column denotes the reconstructed common 3D shapes in a normal viewpoint, where the Red, Green, Blue lines represent left-right, front-end, top-bottom directions, and other directions are represented by the Black lines. Then, we show three samples from each object in three double-columns. In each double-column, the first column is the reconstructed structures annotated in the corresponding 2D images; the second column is the heatmaps, where the depths are represented by different colors (best view in color).

Table 6 Non-Rigid Deformation with Hourglass Input. The mean *shape* and *rotation* errors for *aeroplane, bus, car, sofa, train*. The Roman numerals indicates the index of subtypes for the mean shape error, and mRE is short for the mean rotation error. EP, PF, Sym-EP, Sym-PF are short for EM-PPCA [44], PriorFree [12,13], Sym-EM-PPCA, Sym-PriorFree, respectively.

	aeroplane								**bus**				
	I	II	III	IV	V	VI	VII	mRE	I	II	III	IV	V
EP	0.20	0.53	0.46	**0.42**	0.49	**0.45**	**0.39**	**0.25**	0.51	0.44	0.61	0.47	0.94
PF	0.64	0.73	0.97	0.84	0.77	0.64	0.89	0.42	1.92	1.80	1.76	1.75	1.97
Sym-EP	**0.20**	**0.50**	**0.45**	**0.42**	**0.47**	**0.45**	0.40	**0.25**	**0.26**	**0.20**	**0.38**	**0.28**	**0.70**
Sym-PF	0.45	0.67	0.77	0.68	0.69	0.53	0.59	0.37	1.23	1.13	0.84	1.00	0.98

	bus		**car**										
	VI	mRE	I	II	III	IV	V	VI	VII	VIII	IX	X	mRE
EP	0.70	0.22	0.35	0.23	0.23	0.30	0.22	0.21	0.34	0.61	**0.18**	0.47	0.16
PF	2.01	0.22	1.42	1.02	1.22	1.09	1.09	1.00	1.24	1.25	1.11	1.12	0.41
Sym-EP	**0.51**	**0.13**	**0.29**	**0.18**	**0.17**	**0.27**	**0.20**	**0.19**	**0.34**	**0.61**	0.19	**0.45**	**0.15**
Sym-PF	1.45	0.48	0.78	0.47	0.58	0.74	0.67	0.50	0.68	0.83	0.64	0.79	0.25

	sofa						**train**				
	I	II	III	IV	V	mRE	I	II	III	IV	mRE
EP	0.63	0.58	**0.56**	**0.26**	0.71	**0.43**	**0.53**	0.83	0.56	1.08	0.47
PF	0.70	**0.44**	0.61	0.34	1.02	0.47	1.31	2.12	1.32	1.94	0.65
Sym-EP	**0.59**	0.65	**0.56**	0.32	**0.65**	**0.43**	0.61	**0.63**	**0.43**	**0.94**	**0.45**
Sym-PF	1.38	0.95	1.04	0.47	1.21	0.69	0.83	0.99	0.96	1.21	0.59

10.2 Experiments on Multiple Images with Non-Rigid Deformation

We carry out the experiments on the non-rigid SfM problem using 2D keypoints localized by the stacked hourglass network. The experimental setup is the same as that in Sect. 9.3, except that we do not augment the images by left-right flipping.

The *mean rotation and shape errors* in Table 6 demonstrate that the results for the symmetry algorithms (*i.e.* Sym-EM-PPCA and Sym-PriorFree) are better than

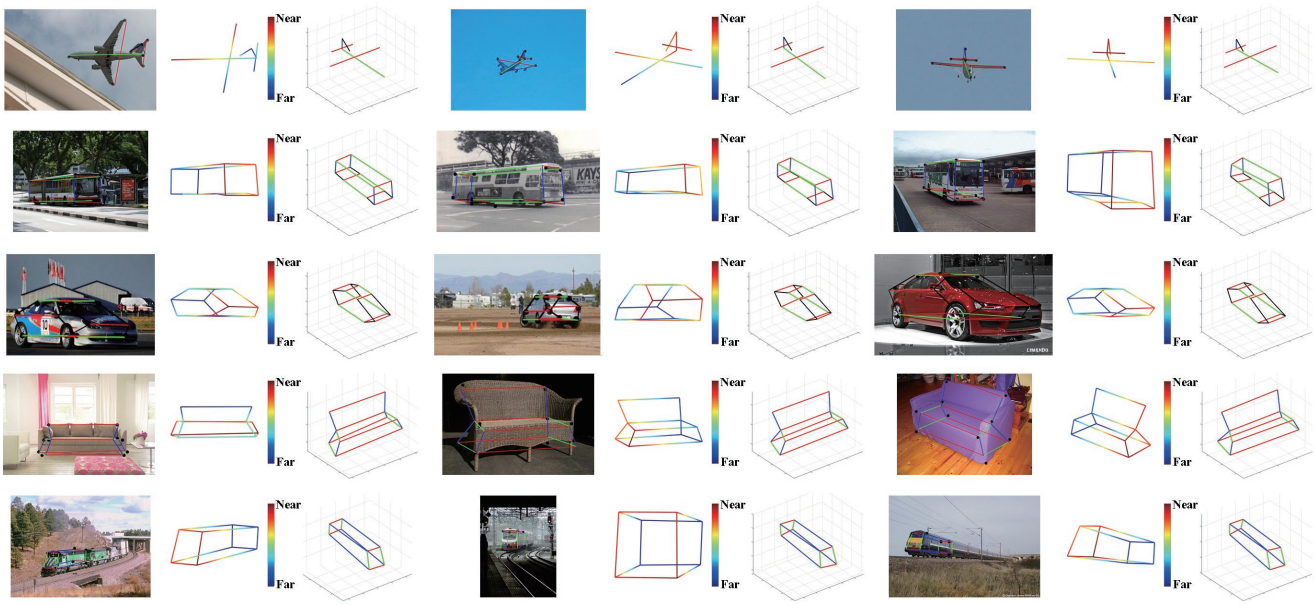

Fig. 8 The illustrations for 3D shapes and viewpoints estimated by Sym-EM-PPCA for *aeroplane, bus, car, sofa,* and *train*. Then, we show three samples from each object in three column-triplets. In each column-triplet, the first column is the reconstructed structures annotated in the corresponding 2D images; the second column is the heatmaps, where the depths are represented by different colors; the third column is a normal view of the reconstructed 3D structures, where the Red, Green, Blue lines represent left-right, front-end, top-bottom directions, and other directions are represented by the Black lines (best view in color).

their non-symmetry counterpart (*i.e.* EM-PPCA and PriorFree). It can also be seen that the performance of the both PriorFree methods (PriorFree and Sym-PriorFree) are generally lower than the EM-PPCA methods (*i.e.* EM-PPCA and Sym-EM-PPCA), the reasons have been discussed in Sect. 9.3. In addition, some results in Table 6 are better than those in Table 3 with manually labeled 2D annotations (though they are not directly comparable), the reasons are the same as those discussed in Sect. 10.1. Some reconstruction results from the Sym-EM-PPCA method are illustrated in Fig. 8.

The non-rigid SfM is not likely to fail to match the reconstructed structures and the 2D keypoints. This is because that the non-rigid SfM reconstruct one 3D structure for each image. But on the other hand, the non-rigid SfM may fail when the deformation among the input images are too severe. The other failure cases are shared between the rigid and non-rigid SfM algorithms: i) imprecise 2D annotations, ii) the input 2D annotations are co-planar, iii) the violation of the orthographic camera assumption (*e.g.* when the object is too close to the camera).

11 Conclusion

Symmetry and Manhattan properties are typically possessed by man-made objects [25, 39]. This paper shows that symmetry, Manhattan and multiple images can be combined together to obtain good quality performance for 3D structure estimation. We first show that symmetry can be exploited by a change of coordinates which decompose the problem into estimating different components of the 3D structure.

For the single image case, we show that symmetry can be exploited, using our change of coordinates, provided Manhattan is used to estimate the camera parameters. This method is successful but has problems when many keypoints are missing due to occlusion, which frequently happens.

Hence we concentrate on multiple images. We show that we can develop new factorization methods which minimize energy functions (which assume that all the keypoints are observed) exploiting symmetry by changing coordinates. These methods require identifying the novel gauge ambiguities, inherent to factorization methods, and identifying strategies to solve for them. But these approaches are limited because they do not work if some of the keypoints are missing due to occlusion. We refer to these energy functions, which assume all the keypoints are observed, as surrogate energy functions.

To address the missing keypoints, we specify new energy functions which take into account that partial keypoints are observed. We can perform coordinate descent on these energy functions, but without good ini-

tialization, the coordinate descent will not converge to good results. So instead we develop methods for initializing and updating the missing keypoints in order to give good starting points for coordinate descent. More specifically, we provide a simple method to initialize the missing keypoints, and use factorization to minimize the surrogate energy functions using the initialized, or estimated, keypoints in order to obtain the 3D structure and the camera viewpoints.

We develop algorithms for the rigid and non-rigid cases. For the rigid case, our algorithm is based on the classic SVD methods modified to deal with missing keypoints. We provide two algorithms for the non-rigid case. The first is based on the prior-free methods of [12, 13] and the second is based on the work of [44].

Our experiments are carried out on Pascal3D+ dataset [47]. They show that our single image reconstruction method achieves good quality results provided symmetry and Manhattan can be identified with suitable keypoints. The results also show that our symmetric rigid structure from motion method, Sym-RSfM, and symmetric non-rigid structure from motion methods, Sym-EM-PPCA and Sym-PriorFree, all almost always outperform the corresponding non-symmetric state-of-the-art methods. To test the robustness of our method, we add Gaussian noise to the keypoint locations and show that our symmetric methods still perform well (as described in the supplementary material).

We also extend our approach to natural images by using an hourglass CNN [36] to localize the keypoints, specify their semantic meanings (i.e. left wheel and right wheel), and output the keypoint pairs. This complete system works very well on the Pascal3D+ dataset, partly because the hourglass gives good initialization for the missing keypoints.

Our work can be extended in several directions. Firstly, all of our current methods use an orthographic or weak-perspective (*i.e.* orthographic plus a scaling) camera model. We expect to obtain better performance if we extend to a full perspective model, particularly for objects which are very close to the camera lens. Secondly, this paper focuses on exploiting the intrinsic symmetry and Manhattan properties of the objects, but uses a relatively simple algorithm to initialize the missing keypoints. We will use more advanced occlusion recovery methods as better initialization, such as [33]. Finally, we will include additional object features, instead of just keypoints, to better improve object 3D structure reconstruction and camera viewpoint estimation.

Acknowledgements We would like to thank Ehsan Jahangiri, Cihang Xie, Weichao Qiu, Xuan Dong, Siyuan Qiao for giving feedbacks on the manuscript. This work was supported by ARO 62250-CS, ONR N00014-15-1-2356, and the NSF award CCF-1317376.

References

1. Agudo, A., Agapito, L., Calvo, B., Montiel, J.: Good vibrations: A modal analysis approach for sequential non-rigid structure from motion. In: CVPR, pp. 1558–1565 (2014)
2. Akhter, I., Sheikh, Y., Khan, S.: In defense of orthonormality constraints for nonrigid structure from motion. In: CVPR (2009)
3. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid structure from motion in trajectory space. In: NIPS (2008)
4. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: A dual representation for nonrigid structure from motion. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(7), 1442–1456 (2011)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
6. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: ECCV (2010)
7. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: CVPR (2000)
8. Ceylan, D., Mitra, N.J., Zheng, Y., Pauly, M.: Coupled structure-from-motion and 3d symmetry detection for urban facades. ACM Transactions on Graphics **33** (2014)
9. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS, pp. 1736–1744 (2014)
10. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: ICCV (1999)
11. Coughlan, J.M., Yuille, A.L.: Manhattan world: Orientation and outlier detection by bayesian inference. Neural Computation **15**(5), 1063–1088 (2003)
12. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. In: CVPR (2012)
13. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. International Journal of Computer Vision **107**, 101–122 (2014)
14. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR (2009)
15. Gao, Y., Yuille, A.L.: Symmetry non-rigid structure from motion for category-specific object structure estimation. In: ECCV (2016)
16. Gao, Y., Yuille, A.L.: Exploiting symmetry and/or manhattan properties for 3d object structure estimation from single and multiple images. In: IEEE International Conference on Computer Vision and Pattern Recognition (2017)
17. Gordon, G.G.: Shape from symmetry. In: Proc. SPIE (1990)
18. Gotardo, P., Martinez, A.: Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 2051–2065 (2011)
19. Grossmann, E., Ortin, D., Santos-Victor, J.: Single and multi-view reconstruction of structured scenes. In: ACCV (2002)

20. Grossmann, E., Santos-Victor, J.: Maximum likehood 3d reconstruction from one or more images under geometric constraints. In: BMVC (2002)
21. Grossmann, E., Santos-Victor, J.: Least-squares 3d reconstruction from one or more views and geometric clues. Computer Vision and Image Understanding **99**(2), 151–174 (2005)
22. Hamsici, O.C., Gotardo, P.F., Martinez, A.M.: Learning spatially-smooth mappings in non-rigid structure from motion. In: ECCV, pp. 260–273 (2012)
23. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press (2004)
24. Hong, J.H., Fitzgibbon, A.: Secrets of matrix factorization: Approximations, numerics, manifold optimization and random restarts. In: ICCV (2015)
25. Hong, W., Yang, A.Y., Huang, K., Ma, Y.: On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image. International Journal of Computer Vision **60**, 241–265 (2004)
26. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: CVPR (2015)
27. Kontsevich, L.L.: Pairwise comparison technique: a simple solution for depth reconstruction. JOSA A **10**(6), 1129–1135 (1993)
28. Kontsevich, L.L., Kontsevich, M.L., Shen, A.K.: Two algorithms for reconstructing shapes. Optoelectronics, Instrumentation and Data Processing **5**, 76–81 (1987)
29. Li, Y., Pizlo, Z.: Reconstruction of shapes of 3d symmetric objects by using planarity and compactness constraints. In: Proc. of SPIE-IS&T Electronic Imaging (2007)
30. Ma, J., Zhao, J., Ma, Y., Tian, J.: Non-rigid visible and infrared face registration via regularized gaussian fields criterion. Pattern Recognition **48**(3), 772–784 (2015)
31. Ma, J., Zhao, J., Tian, J., Bai, X., Tu, Z.: Regularized vector field learning with sparse approximation for mismatch removal. Pattern Recognition **46**(12), 3519–3532 (2013)
32. Ma, J., Zhao, J., Tian, J., Tu, Z., Yuille, A.L.: Robust estimation of nonrigid transformation for point set registration. In: CVPR, pp. 2147–2154 (2013)
33. Marques, M., Costeira, J.: Estimating 3d shape from degenerate sequences with missing data. Computer Vision and Image Understanding **113**(2), 261–272 (2009)
34. Morris, D.D., Kanatani, K., Kanade, T.: Gauge fixing for accurate 3d estimation. In: CVPR (2001)
35. Mukherjee, D.P., Zisserman, A., Brady, M.: Shape from symmetry: Detecting and exploiting symmetry in affine images. Philosophical Transactions: Physical Sciences and Engineering **351**, 77–106 (1995)
36. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, pp. 483–499. Springer (2016)
37. Olsen, S.I., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. Journal of Mathematical Imaging and Vision **31**(2-3), 233–244 (2008)
38. Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K.: 6-dof object pose from semantic keypoints. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, pp. 2011–2018. IEEE (2017)
39. Rosen, J.: Symmetry discovered: Concepts and applications in nature and science. Dover Publications (2011)
40. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**, 1–10 (1966)
41. Thrun, S., Wegbreit, B.: Shape from symmetry. In: ICCV (2005)
42. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9**(2), 137–154 (1992)
43. Torresani, L., Hertzmann, A., Bregler, C.: Learning non-rigid 3d shape from 2d motion. In: NIPS (2003)
44. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**, 878–892 (2008)
45. Vetter, T., Poggio, T.: Symmetric 3d objects are an easy case for 2d object recognition. Spatial Vision **8**, 443–453 (1994)
46. Vicente, S., Carreira, J., Agapito, L., Batista, J.: Reconstructing pascal voc. In: CVPR (2014)
47. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: WACV (2014)
48. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to nonrigid shape and motion recovery. In: ECCV (2004)