

Article

3D Reconstruction of Space Objects from Multi-Views by a Visible Sensor

Haopeng Zhang ^{1,2,*}, Quanmao Wei ^{1,2} and Zhiguo Jiang ^{1,2,*}

¹ Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China; weiqm@buaa.edu.cn

² Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China

* Correspondence: zhanghaopeng@buaa.edu.cn (H.Z.); jiangzg@buaa.edu.cn (Z.J.); Tel.: +86-10-8233-8061 (H.Z. & Z.J.)

Received: 1 June 2017; Accepted: 20 July 2017; Published: 22 July 2017

Abstract: In this paper, a novel 3D reconstruction framework is proposed to recover the 3D structural model of a space object from its multi-view images captured by a visible sensor. Given an image sequence, this framework first estimates the relative camera poses and recovers the depths of the surface points by the structure from motion (SFM) method, then the patch-based multi-view stereo (PMVS) algorithm is utilized to generate a dense 3D point cloud. To resolve the wrong matches arising from the symmetric structure and repeated textures of space objects, a new strategy is introduced, in which images are added to SFM in imaging order. Meanwhile, a refining process exploiting the structural prior knowledge that most sub-components of artificial space objects are composed of basic geometric shapes is proposed and applied to the recovered point cloud. The proposed reconstruction framework is tested on both simulated image datasets and real image datasets. Experimental results illustrate that the recovered point cloud models of space objects are accurate and have a complete coverage of the surface. Moreover, outliers and points with severe noise are effectively filtered out by the refinement, resulting in an distinct improvement of the structure and visualization of the recovered points.

Keywords: space object; 3D structural model; 3D reconstruction; structure from motion; point cloud refinement

1. Introduction

Detecting, tracking, cataloging and identifying man-made objects orbiting Earth comprise one of the fundamental requirements of space surveillance [1]. More and more states are pursuing such space surveillance systems to develop their space surveillance capabilities [2]. In recent years, space-based surveillance systems [3–5] and high-performance optical imaging sensors have been rapidly developed. Visible sensors used by space-based surveillance systems can avoid the impact of atmospheric turbulence, which severely influences traditional ground-based systems. Meanwhile, space-based surveillance systems can also provide higher spatial resolution image data at a closer distance.

Object categorization, recognition and pose estimation of space objects are the main tasks in the fields of space exploitation and surveillance. In terms of pose estimation, Zhang et al. [6] presented a vision-based method. Zhang et al. [7] proposed an improved pose estimation algorithm based on 2D-3D correspondences between the input image and the 3D model of space objects. For space object recognition, different features and clustering algorithms were studied [8–11]. To handle the tasks of both recognition and pose estimation in one vision-based framework, methods based on kernel regression [12] and homeomorphic manifold analysis [13] were proposed. However, all of these methods require, directly or indirectly, the prior 3D models of space objects. For example, the method of [7] needs the 3D model to provide 2D-3D correspondences, and those of [6,8–13] need images from

multiple or full viewpoints, which are difficult to access without prior models of the targets. Thus, obtaining 3D models is important for vision-based tasks in space surveillance.

In this paper, we address the problem of recovering space object models by reconstruction from images captured by space-based visible sensors. Compared to model generation by laser scanning, image-based 3D reconstruction is economical and convenient. Moreover, it is also a non-contact and noninvasive measurement, which is suitable for space missions, especially for non-cooperative targets. On the one hand, the recovered space object models and full-viewpoint images generated with the recovered models are helpful for the recognition and pose estimation. Furthermore, the recovered 3D models could lead to a better result for automatic measurement and pose estimation of the sub-components, as these 3D models do not contain the perspective projection, which must be taken into account when handling two-dimensional images. On the other hand, the recovered 3D models have a great application value for space missions, such as autonomous rendezvous and docking, collision avoidance and on-orbit self-serving. Moreover, the recovered models could further reveal the functional characteristics of space objects, which are significant for space situational awareness.

Multiple view geometry has developed rapidly since the 1990s [14], and many image-based reconstruction systems have been proposed [15–23]. However, there are few works that focus on the 3D structural model reconstruction of space objects, which is indeed the goal of this paper. This paper is a follow-up to the work in [24]. In this paper, traditional multi-view-image-based reconstruction pipelines are applied, including image feature point extracting and matching, sparse reconstruction by the structure from motion (SFM) method [14,22,23] and dense reconstruction by the patch-based multi-view stereo (PMVS) algorithm [25]. A special modification of the strategy of adding new images during SFM is introduced to resolve the wrong matches that arise from the symmetric structure and the repeating textures of space objects. Additionally, a refinement method exploiting the structural characteristic that most sub-components of space objects are made out of basic geometric shapes is proposed to improve the visualization of the recovered point cloud model. The experimental results on both simulated image datasets and real image datasets have demonstrated the reconstruction ability and accuracy of this framework.

This paper is organized as follows. A brief introduction of the framework is given in Section 2. Details of the reconstruction and point cloud refining method are described in Sections 3 and 4, respectively. Experimental results are shown in Section 5. At last, Section 6 provides the conclusion.

2. Overview of the Framework

The reconstruction framework for space objects consists of two parts, i.e., reconstruction of the 3D structural model from multi-view images and refinement of the reconstructed point cloud model, as illustrated in Figure 1.

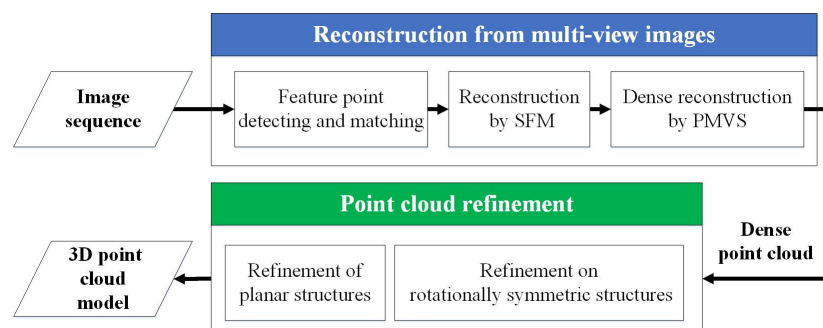


Figure 1. Framework for the reconstruction of space objects. PMVS, patch-based multi-view stereo.

The theoretical principles of 3D reconstruction are well defined, and general reconstruction pipelines, especially the SFM, are also widely used. Therefore, to recover the 3D structural model from

multi-view images, our framework employs a general feature point-based multi-view reconstruction procedure, which consists of three steps:

- (a) Detect feature points in each image and match the feature points between each pair of images.
- (b) Estimate the relative poses of each view and recover the depths of the detected points, i.e., the SFM step.
- (c) Taking the results of SFM as inputs, use the PMVS algorithm [25] to generate a dense 3D point cloud.

Due to the structure symmetry and repeated textures of space objects, there might be enormous error matches between symmetrical viewpoints, which meet the geometrical conditions while disagreeing with the actual imaging configurations. These error matches would further lead to overlap and failure of the reconstruction. To handle such mismatches, a special modification is made on the strategy of adding new images during the SFM step. This paper assumes that the space objects are mainly composed of planar solar wings and a cuboid/cylinder main body, and the datasets used in our experiments basically satisfy this assumption. Small and thin functional/non-functional components, such as antennas, imaging sensors, nozzles and trusses, are all ignored (actually treated as outliers) in our refinement processing. However, such simplified general models are still sufficient for most space applications.

3. 3D Reconstruction from Multi-View Images

In this paper, a general linear pinhole camera model is adopted. The intrinsic parameter matrix \mathbf{K} of the camera is modeled as:

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where f denotes the focal length in pixels, (c_x, c_y) represents the image coordinate of the principal point, i.e., the point where the principal axis of the camera intersects the image plane. Since there is little distortion in our simulated images, a single ideal camera with a fixed focal length is adopted rather than a more complex distorted one. Due to the development of calibration methods, it is easy to calibrate the intrinsic parameters with high precision. Thus, a given intrinsic parameter matrix \mathbf{K} , which could be calibrated in advance, is required to simplify the problem.

3.1. Feature Point Detecting and Matching

The scale-invariant feature transform (SIFT) keypoints [26] are extracted for each image as feature points because of their stability and invariance to image transformations. In addition to the keypoint locations, the SIFT detector also provides a 128-dimensional local descriptor for each keypoint, which can be used for the primary match with an approximate nearest neighbor search method. A point ${}^m p_i$ (the i -th point in the m -th image) in image F_m is matched to a point ${}^n p_j$ (the j -th point in the n -th image) in image F_n when two conditions are satisfied: (a) among all of the feature points in F_n , the descriptor of ${}^n p_j$ has the closest distance (e.g., the Euclidean distance) d_{1st} to that of ${}^m p_i$; (b) d_{1st} is much smaller than the second closest distance d_{2nd} , i.e., $d_{1st} < \lambda d_{2nd}$, $0 < \lambda < 1$. Then, for point matches $\{{}^m p_i, {}^n p_j\}$ between image pair F_m and F_n , to remove the outliers, the random sample consensus (RANSAC) algorithm [27] is used, in which the epipolar constraint is employed, i.e.,

$${}^m p_i \mathbf{E} {}^n p_j = 0, \quad (2)$$

where ${}^m p_i$ and ${}^n p_j$ denote the homogeneous normalized coordinate vectors of a point pair between image frame F_m and F_n . Here, the normalization means conversion from image coordinate to camera coordinate, i.e., $p \equiv \hat{p} = \mathbf{K}^{-1} \begin{bmatrix} x & y & 1 \end{bmatrix}^T$, where (x, y) is the image coordinate. $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$ is the

essential matrix, which can be parameterized by rotation matrix \mathbf{R} and translation vector \mathbf{t} , and $[\bullet]_{\times}$ denotes the skew-symmetric matrix, such that $[\mathbf{a}]_{\times} \mathbf{x} = \mathbf{a} \times \mathbf{x}$.

The essential matrix \mathbf{E} , with a non-zero scale factor, is simultaneously estimated by a robust method, such as the eight-point algorithm [28], during the RANSAC iteration. A more accurate estimation of \mathbf{E} can be achieved by non-linear refinement after the iteration.

3.2. Structure from Motion

Structure from motion begins with the relative camera pose estimation of an initial image pair. Such an initial image pair is selected with certain rules, for instance the image pair that has the largest number of point matches and a large baseline, as proposed in [22]. Assume F_0 and F_1 are the initial image pair. Since the essential matrix \mathbf{E}_1 between F_0 and F_1 has already been found in Section 3.1, the rotation matrix \mathbf{R}_1 and translation vector \mathbf{t}_1 of F_1 , referring to the camera coordinate of F_0 , can be further extracted from \mathbf{E}_1 by SVD-based techniques. Then, the 3D locations of the matched point pairs between F_0 and F_1 can be recovered by triangulation, yielding an initial 3D point set M_1 . To refine both the estimated camera pose (i.e., the rotation matrix \mathbf{R} and translation vector \mathbf{t}) and the recovered 3D point set jointly, bundle adjustment is used. The goal of bundle adjustment is to minimize the reprojection errors through optimizing the positions of both the cameras and the observed points, i.e.,

$$\min_{\mathbf{P}_j \in M_n, \mathbf{R}_n, \mathbf{t}_n} \sum_{i=0}^n \sum_{j=0}^{m_i-1} \| {}^i p_j - [\mathbf{R}_i | \mathbf{t}_i] \mathbf{P}_j \|^2, \quad (3)$$

where $M_n = \{\mathbf{P}_j\}$ is the last updated 3D point set with a size of m_i . \mathbf{R}_i and \mathbf{t}_i are the rotation matrix and translation vector, which indicate the camera pose of image F_i . ${}^i p_j$ is the projection point of the recovered 3D point \mathbf{P}_j in image F_i . Here, $n = 1$ for the initial pair.

Next, the image that observes the largest number of recovered points in M_{n-1} is added into the n -th iteration. Let M_{n-1} be the recovered 3D point set after adding the $(n-1)$ -th image F_{n-1} . The rotation matrix \mathbf{R}_n and translation vector \mathbf{t}_n are then estimated from the 2D-3D point correspondences between F_n and M_{n-1} . Next, point matches between the new added image F_n and all of the previous images F_m ($m = 0, 1, \dots, n-1$), but having no corresponding point in M_{n-1} , are recovered, resulting in the increase of size from M_{n-1} to M_n , i.e., a more complete reconstructed structure. At last, the bundle adjustment is applied to optimize \mathbf{R}_n , \mathbf{t}_n and M_n . This procedure is repeated until there is no image remaining.

Artificial space objects, in common with most industrial products, are designed with symmetric structures and have repeating textures. These characteristics might cause point matches between images that are seen from opposing viewpoints, as shown in Figure 2a. Then, a wrong camera pose would be estimated from such unexpected matches and further influence the rest of the estimations of camera poses. As shown in Figure 2b, the recovered camera viewpoints are approximately located on a half of circle, while in fact, the test images are taken from viewpoints uniformly located on a full circle. The final point cloud recovered is incorrect, as shown in Figure 2c. To solve this problem, we modify the original strategy where only the number of point correspondences is considered. Since the inputs in our cases are image sequences, i.e., the images are well sorted in imaging order, the overlaps between neighboring images should ensure enough point matches. Therefore, to avoid the unexpected matches between images from opposing viewpoints and ensure enough correct matches between the newly-added image and recovered points, as well, our modified strategy is to add the images only in the order that they are captured. The correct reconstruction result with our modified image adding strategy is shown in Figure 3, where both the viewpoints and structural model are correctly recovered.

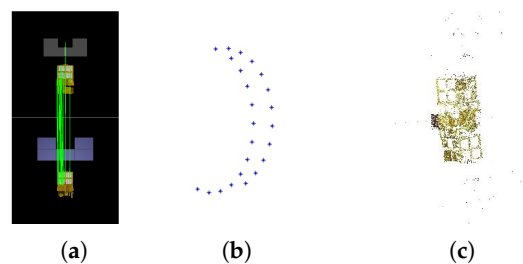


Figure 2. Incorrect reconstruction caused by symmetric structures and repeated textures. (a) Matches; (b) poses; (c) point cloud.

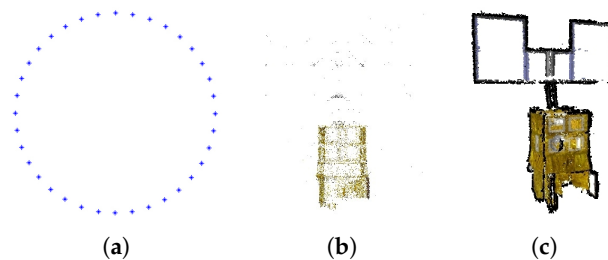


Figure 3. Correct reconstruction with modified image adding strategy. (a) Poses; (b) point cloud (sparse); (c) point cloud (dense).

3.3. Dense Reconstruction

Through SFM, we can estimate the relative camera pose of each image and recover the 3D location of feature points, but such reconstruction of the feature points is not enough to reveal the structure of the target due to the poor surface coverage. Therefore, to improve the reconstruction coverage, the PMVS algorithm [25] is employed. The space object images, in which the backgrounds are not cluttered and the object is usually prominent, are fine inputs for PMVS, as it could be relatively straightforward to extract the contours. After the iterations of the matching, expanding and filtering procedure, PMVS can output a dense set of rectangular patches covering the object surfaces that are visible in the input images. Each patch is defined by a combination of a center point and a normal vector.

4. Point Cloud Refinement

The main body of the space object might be a cuboid or symmetric with respect to its major axis, and the solar panels are always flat and could be treated as rectangles when ignoring their thickness. With this structural prior information that most sub-components are in basic geometric shapes, refinements for planes and rotationally-symmetric structures are introduced respectively. Although the precise refinement operations for these two structures are various, they all follow the same procedure. Specifically, the basic geometric structure is first detected, then points belonging to this structure are discriminated, and finally, an adjustment is done to each point.

4.1. Refinement for Planes

The planes are first detected by Hough transform in 3D space. Assume Π is some 3D plane to be detected, then Π can be formulated as $ax + by + cz + d = 0$, where $(x, y, z, 1) \in \Pi$ is the point on Π in homogeneous format and (a, b, c, d) is the parameter that defines Π . This formulation can also be treated as follows: a point (a, b, c, d) is on a plane defined by $(x, y, z, 1)$, due to the dual relation between the plane parameter and point coordinate. Given series points in Π , the parameter (a, b, c, d) defining Π then can be found as an intersection point of the planes that are defined by these given points. Since (a, b, c, d) is homogeneous, any parameter $(\lambda a, \lambda b, \lambda c, \lambda d)$ ($\lambda \neq 0$) refers to the same plane.

To ensure uniqueness, a , b and c are fixed to one separately in three separate Hough detections, and the other three parameters are discretized and limited in range $[-1, 1]$. Discretization with resolution 0.1 is illustrated as Figure 4, where all of the possible normal vectors are approximately uniformly distributed. The plane parameters of Π then correspond with a discretized point in the parameters space, and the final plane is picked from three candidates obtained by the three Hough detections.

After plane Π is detected, points that belong to the plane structure Π are then discriminated. A point p is discriminated to be on Π when: (a) p is near Π ; and (b) the normal vector angle between p and Π is small. Considering that different planar components might be located on the same plane, e.g, two symmetrically-assembled solar wings that face in the same direction, the region growing algorithm is employed to distinguish such different components. At last, each point discriminated to be on Π is moved to its projection on Π , and the normal vectors of these points are set to be parallel with the normal vector of Π .

The whole procedure of the plane refinement is shown in Algorithm 1. Multiple planes are iteratively detected, until the last detected plane is too small or the number of detected planes exceeds a expected maximum Max_{np} ($Max_{np} = 8$ in our experiments).

Algorithm 1: Refinement for planes.

Input: The original 3D points set X_0 with N_0 elements.
Output: The refined 3D points set X' .
 $Max_{np} \leftarrow$ the number of planes supposed to be in X_0 ;
The number of detected planes $k \leftarrow 0$;
 $X' \leftarrow \emptyset$;
while $k < Max_{np}$ **do**
 $((1, b_1, c_1, d_1), v_1) \leftarrow Hough_{a=1}(X_k)$; // v is the number of votes;
 $((a_2, 1, c_2, d_2), v_2) \leftarrow Hough_{b=1}(X_k)$;
 $((a_3, b_3, 1, d_3), v_3) \leftarrow Hough_{c=1}(X_k)$;
 $q \leftarrow \arg \max_{i \in \{1,2,3\}} v_i$;
 $((a_q, b_q, c_q, d_q), v_q) \leftarrow ((a_q, b_q, c_q, d_q), v_q)$;
 if $v_q/N_k < r$ **then**
 | **Break** from *While*-loop; // the area of the plane might be too small;
 end
 Plane $\Pi_k \leftarrow a_kx + b_ky + c_kz + d_k = 0$;
 for each point p_i **in** X_k **do**
 | $n_i \leftarrow$ the distance between p_i and Π_k ;
 | $\theta_i \leftarrow$ the normal vector angle between p_i and Π_k ;
 | **if** $n_i \leq n$ and $\theta \leq \theta$ **then**
 | p_i is located on Π_k , $p_i \in C_{\Pi_k}$;
 | **end**
 end
 $C_{\Pi_k} \leftarrow$ do Region Grow on C_{Π_k} and pick the cluster with most points;
 for each point p_j **in** C_{Π_k} **do**
 | Move p_j to its projection on Π_k ;
 | Set the normal vector of p_j to be the same as Π_k ;
 end
 $X_{k+1} \leftarrow X_k - C_{\Pi_k}$;
 $N_{k+1} \leftarrow$ size of X_{k+1} ;
 $k \leftarrow k + 1$;
end
Result: $X' \leftarrow X_k + \sum_{i=0}^{k-1} C_{\Pi_k}$.

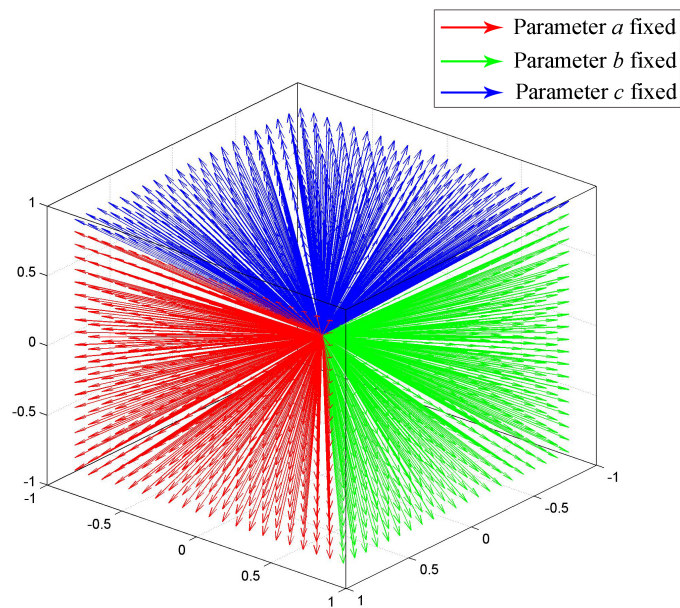


Figure 4. Illustration of normal vectors. All of the possible normal vectors of Π are approximately uniformly distributed in the first octant. Here, the discretized resolution of the plane parameters is 0.1.

4.2. Refinement for Rotationally-Symmetric Structures

Parameterizing a cylinder requires more parameters than a simple plane; this increase of parameter count would result in a geometrical growth of calculation in Hough transformation. Moreover, rotationally-symmetric structures are not limited to the cylinder, namely the parametrization of rotationally-symmetric structures could be complicated and vary with a specific structure. Considering the common characteristic of all rotationally-symmetric structures, consisting of the surface normal always intersecting the symmetry axis, the detection of such rotationally-symmetric structure could be converted into the detection of the symmetric axis from those intersections. The 3D space is first discretized to voxels at a suitable resolution. Next, the voxels are voted based on the surface normals passing through them. Allowing for the existence of intersections of non-surface normals and the error of surface normals, only voxels with enough votes are accepted as the right intersections of surface normals. Then, principal component analysis (PCA) is used to robustly estimate the symmetry axis from these intersections, and the votes are used as weights.

After finding the symmetry axis, the surface points are discriminated preliminarily by the rule that a point is a surface point if the distance between its normal and the symmetry axis is short enough. Then, the surface points are refined, and during the refinement, these points are verified again. First, the space is re-divided into layers with a thickness of Δd along the symmetry axis. Then, the mean radius R_i is calculated for each layer L_i . Referring to R_i , points too far away from or too near to the symmetry axis are removed from surface points, and R_i is recalculated after this verification. At last, each surface point in L_i is moved along the radius direction to keep a distance of R_i from the symmetry axis, and the normals of these points are set so that they will exactly intersect the symmetry axis at the mean position of the original intersections.

The whole refinement procedure for rotationally-symmetric structures is shown in Algorithm 2, and the adjustment of the surface points is shown in Figure 5.

Algorithm 2: Refinement for rotationally-symmetric structures.

Input: The original 3D points set X_0 .
Output: The refined 3D points set X' .
 $X' \leftarrow \emptyset$;
for each point p_i in X_0 **do**
 | Vote for every voxels that the normal of p_i passes through;
end
 $m \leftarrow$ half of the max votes;
 $(u_x, u_y, u_z) \leftarrow$ the mean location of voxels whose votes are no less than m ;
 $(n_x, n_y, n_z) \leftarrow$ the principal component of the distribution of voxels whose votes are no less than m by PCA;
The symmetry axis \mathbf{N} is defined as $[u_x, u_y, u_z]^T + \lambda [n_x, n_y, n_z]^T$;
for each point p_i in X_0 **do**
 | $d_i \leftarrow$ the distance between the normal of p_i and \mathbf{N} ;
 if $d_i \leq d$ **then**
 | Accept p_i as surface point;
 end
end
Divide the space into layers with thickness of Δd along \mathbf{N} ;
for each layer L_i **do**
 for each surface point p_{ij} in L_i **do**
 | $r_{ij} \leftarrow$ distance between p_{ij} and \mathbf{N} ;
 end
 $R_i \leftarrow$ trimmed mean of r_{ij} ;
 for each surface point p_{ij} in L_i **do**
 if $r_{ij} > (1 + \alpha)R_i$ or $r_{ij} < \beta R_i$ **then**
 | Remove p_{ij} from surface points;
 end
 end
 for each verified surface point p_{ij} in L_i **do**
 | $r_{ij} \leftarrow$ distance between p_{ij} and \mathbf{N} ;
 | $l_{ij} \leftarrow$ where the normal of p_{ij} intersects \mathbf{N} ;
 end
 $R_i \leftarrow$ trimmed mean of r_{ij} ;
 $S_i \leftarrow$ trimmed mean of l_{ij} ;
 for each verified point p_{ij} in L_i **do**
 | Adjust p_{ij} so that $\mathbf{r}_{ij} = \frac{R_i}{|\mathbf{r}_{ij}|} \mathbf{r}_{ij}$;
 | Reassign the normal of p_{ij} so that the normal of p_{ij} intersects \mathbf{N} at S_i ;
 end
end
Result: $X' \leftarrow$ refined X_0 .

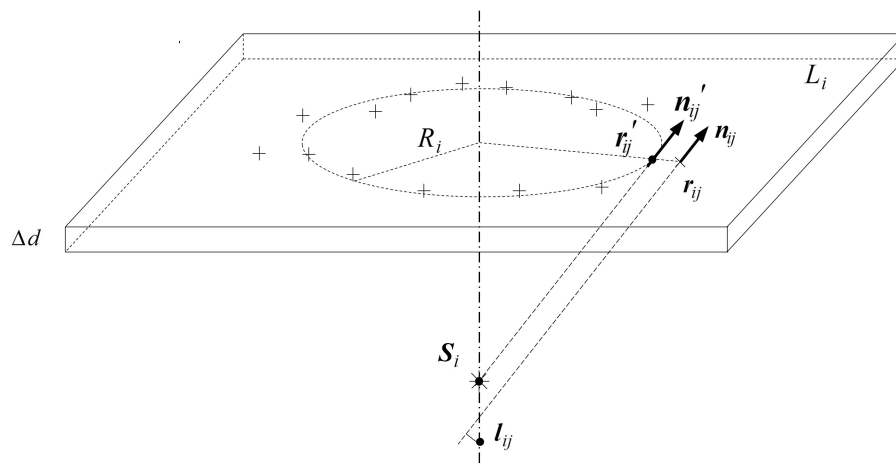


Figure 5. Illustration of point adjustment for rotationally-symmetric structures. For layer L_i , move the point at r_{ij} along the radius direction to r'_{ij} , and set its normal as $n'_{ij} = S_i - r'_{ij}$, where S_i is the trimmed mean location of $\{l_{ij}\}$.

5. Experiments and Analyses

In this section, the proposed space object reconstruction framework is tested on the datasets in Section 5.1. Note that the reconstruction procedure in our framework is implemented based on Bundler [22], where the image addition mechanism is modified. To quantitatively verify the feasibility and accuracy of the refinement method, experiments are conducted on both simulated images and real images.

5.1. Data Collection

The image data used in the experiments include the following image sets:

1. Images of spacecraft Shenzhou-6 and Tiangong-1 from the ground imaging simulation experiment. In the ground imaging simulation experiment, a dark room covered with cloth absorbing light inside and cloth reflecting light outside is established to simulate the space environment. Parallel light is used to simulate the sun light. Camera is fixed pointing at the scaled model of the space object, which is put on a one-degree-of-freedom turntable and rotated along with the turntable. One frame is taken every 10° of rotation of the turntable. The supporter and turntable in images are erased later, as shown in Figure 6.
2. Rendering images of CAD models of a box and a cylinder: The virtual camera is fixed pointing at the model, and the geometric models are attached with different textures. One hundred frames are rendered during a 360° rotation of the model, i.e., one frame is taken every 3.6° the model rotates, as shown in Figure 6.
3. Real images taken from two packages separately: The chosen packages are a packing box of a printer cartridge and a packing canister for badminton, respectively, representing a standard box and a standard cylinder. The package is put on the one-degree-of-freedom turntable and rotated along with it, and the camera is fixed pointing at the package. One image is taken every 2° of rotation. The supporter and turntable in the images are erased later, as well, as shown in Figure 6.

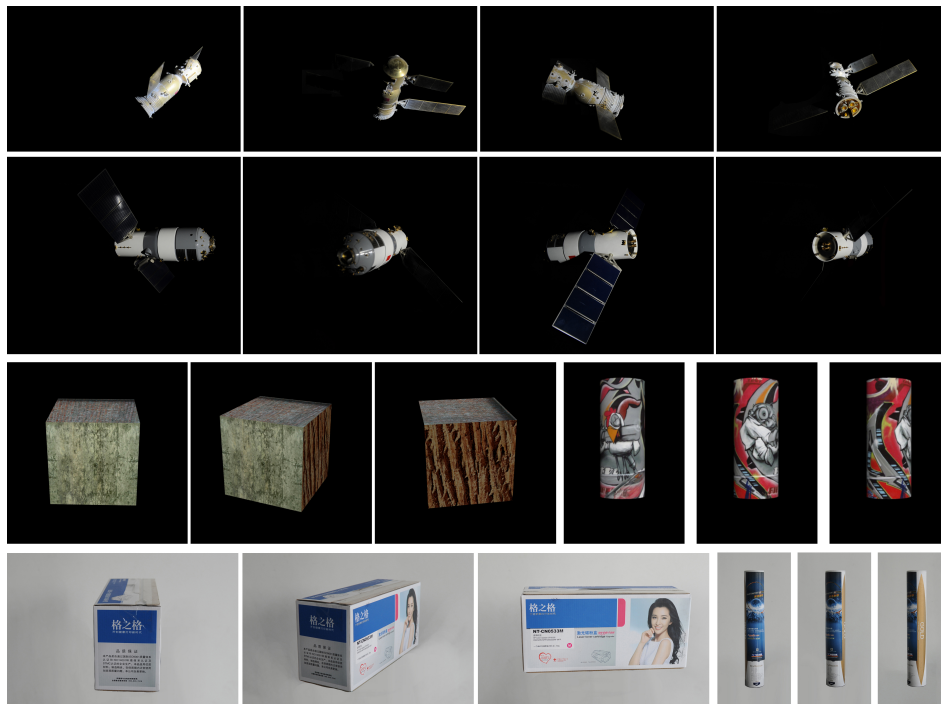


Figure 6. Samples of image data used in our experiments. From top to bottom: image samples of Shenzhou-6 (**the first row**) and Tiangong-1 (**the second row**), rendered images of two 3D CAD models (**the third row**) and real images taken from two actual packages (**the fourth row**).

5.2. Parameter Determination for Point Cloud Refinement

In the point cloud refinement, several thresholds are used, including distance thresholds and angle thresholds. Since the reconstructed 3D point cloud model is up to a scale factor in size, the distance thresholds are relative as well. Thus, to determine such distance thresholds, factor a , which can be regarded as a measurement unit, is first defined based on the overall size of the reconstructed point cloud. In our experiments, a is adaptively defined as 1% of D_3 , where D_3 is the dimension along the last eigenvector ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) of the covariance tensor of these 3D points. The discretization resolutions of the 3D space for the detection of both planes and rotationally-symmetric structures are a . A set of a -based thresholds that can contribute to a fine performance are determined by experience, as listed in Table 1.

Among these threshold parameters, the verification thresholds, i.e., n and θ for planes, and d for rotationally-symmetric structures, are the primary factors that affect the results, since they are used to decide whether a point is a surface point. Experiments are conducted based on the rendered images of CAD models to evaluate the influence of these verification thresholds. Results of plane detection with different verification thresholds n and θ are shown in Figure 7. Small n has little robustness to the distance deviation and is more likely to result in repeated detection of the same plane, while small θ has little robustness to the normal deviation. Thus, a moderate combination of $n = 2a$ and $\theta = 30^\circ$ is used. Figure 8 shows the results of rotationally-symmetric structure detection with different verification thresholds d . Similarly, small d has little robustness to the normal deviation, while large d would abate the filtering effects. Thus, an appropriate small $d = 10a$ is finally used.

Table 1. Definition of thresholds used in point cloud refinement.

Geometry	Thresholds Definition	Value
Plane	r Percentage threshold to decide if the proposal plane is big enough to be a plane component.	0.015~0.02
	n Distance threshold to decide if a point is on the plane.	$2a$
	θ Angle threshold to decide if a point fits the plane.	30°
Rotationally-Symmetric Structure	m Count threshold to decide if the proposal intersection is a intersection of the surface normals.	$Max/2$
	d Distance threshold between the normal a point and the symmetry axis to decide if the point is a surface point.	$10a$
	Δd The thickness resolution of layers divided along the axis.	$a/2$
	α The max radius $(1 + \alpha)R$ for a surface point in consideration of errors.	0.5
	β The min radius βR for a surface point in consideration of errors.	0.2

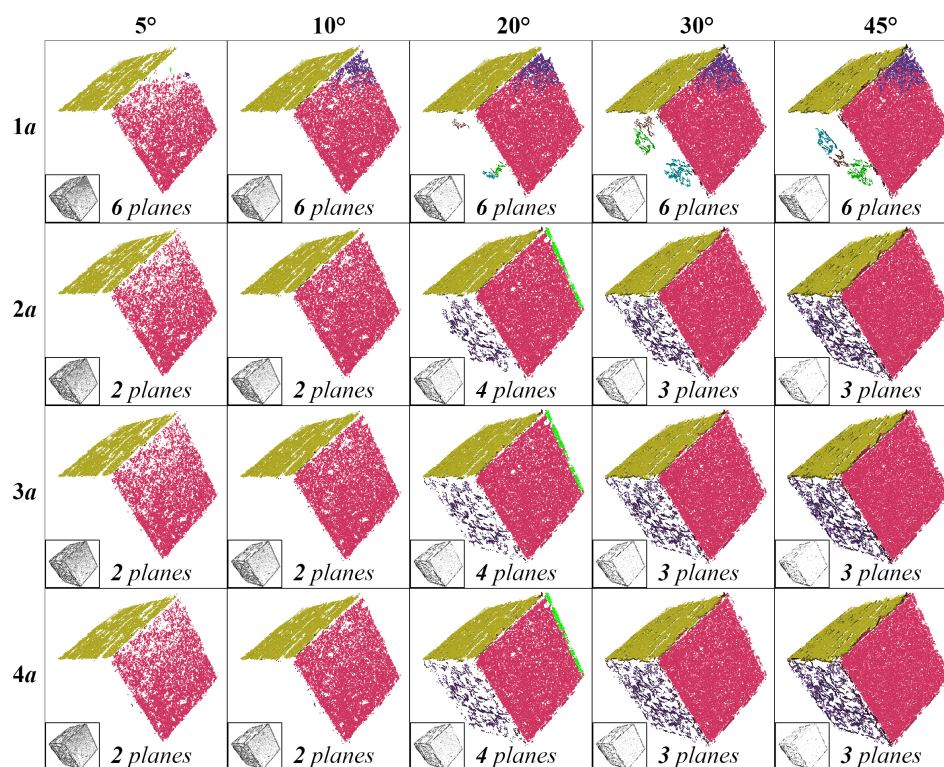


Figure 7. Results of plane detection with different verification thresholds n (rows) and θ (columns). Max_{np} in Algorithm 1 is six. Multiple planes are detected with different colors; point clouds at the lower left corner are the points filtered out.

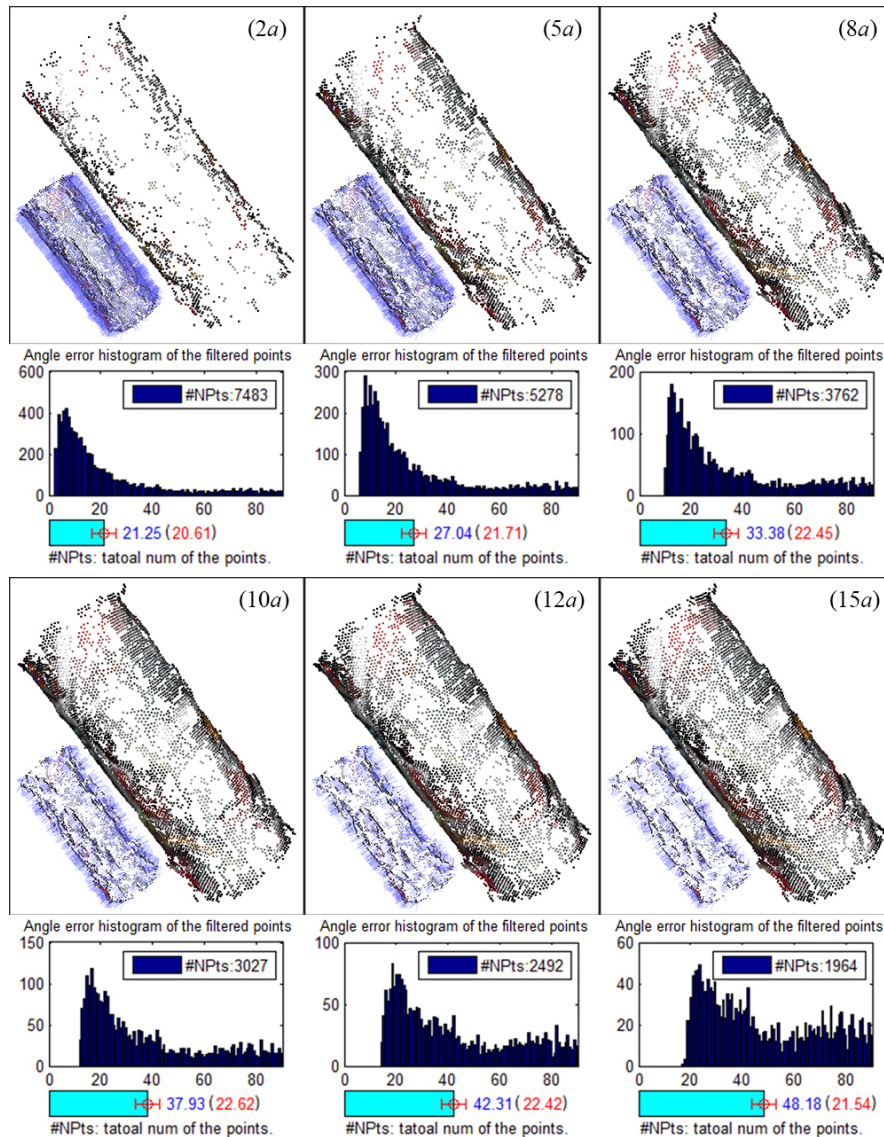


Figure 8. Results of rotationally-symmetric structure detection with different verification threshold d . Point clouds at the lower left corner, which are rendered with normals, are the points filtered out. The bar chart below the point cloud shows the histogram of the angle error of these filtered points, along with a horizontal bar indicating the average and RMS.

5.3. Reconstruction of Objects with Basic Geometry

The results of reconstruction and refinement for the box model and packing box are displayed in Figure 9a,b, where the faces that are visible in the input images are recovered, and the points that have a greater error are filtered out by refinement. Note that, since the simulated images lack good views of the upper face (i.e., viewing above the model), the upper faces of both the box model and packing box are relatively poorly reconstructed. Meanwhile, uniform areas of the packing box cannot be recovered because no stable feature points can be extracted in such areas. Quantitative analysis is shown in Table 2, where the number of recovered points is counted, and the normal vectors of surfaces are calculated, as well. Angles between the normal vectors are $\{90.6^\circ, 90.6^\circ, 90.5^\circ\}$ and $\{89.1^\circ\}$, which are close to the ground truth of 90° .

Results of reconstruction and refinement for the cylinder model and packing canister are displayed in Figure 9c,d. An incorrect recovered part can be found at one end of the packing canister, as shown in Figure 10, that is exactly where the supporter and turntable are erased. The ratio of the length to the

radius is calculated for quantitative analysis. The length of the cylinder is directly measured from the outline of the reconstructed points, which is generated with the mean radius R in each layer along the symmetry axis L , as shown in Figure 11. The mean value of the outlines is calculated as the final radius. The final measurement results are shown in Table 3; deviations from their ground truths are all less than 5%.

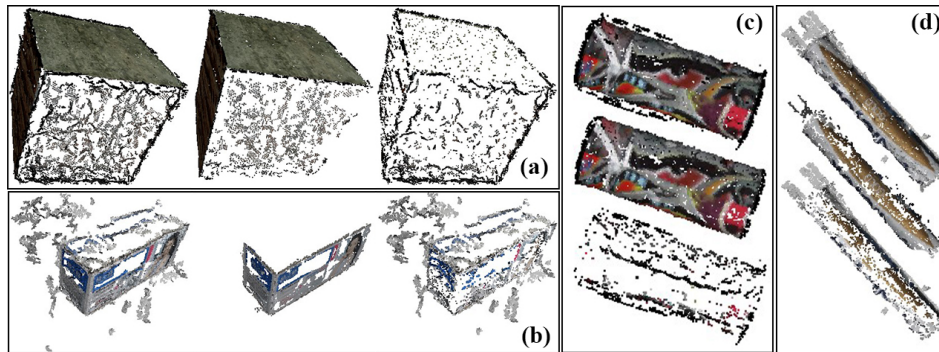


Figure 9. Results of reconstruction and refinement for the box model (a), the packing box (b), the cylinder model (c) and the packing canister (d). From left to right (top to bottom): the recovered point cloud, point cloud after being refined and points filtered by refinement.

Table 2. Analysis for reconstruction of box objects.

Object	N_1 *	N_2 **	Normal Vector of Surfaces
Box Model	41,337	32,761	(0.09, 0, 1)
			(1, -0.01, -0.1)
			(0, 1, -0.01)
Packing Box	27,173	15,266	(1, -0.04, -0.94)
			(0.97, -0.01, 1)

* Number of recovered point clouds; ** Number of point clouds after being refined.

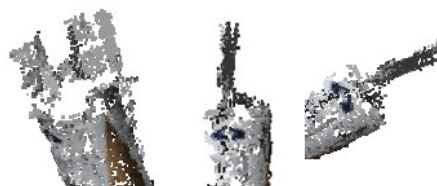


Figure 10. Incorrect reconstruction caused by the erasure of the supporter and turntable.

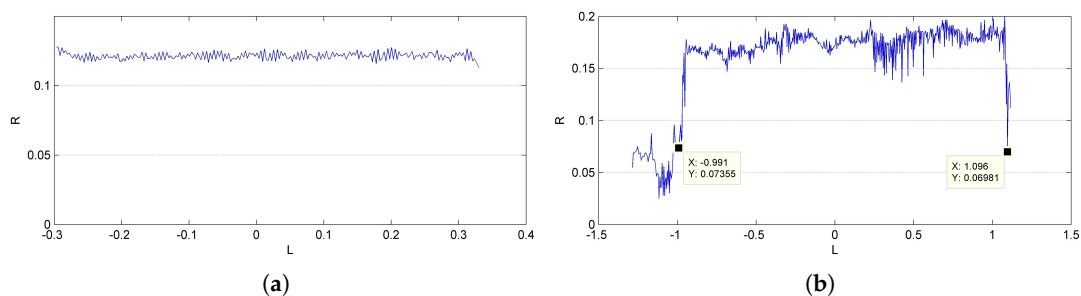


Figure 11. Outline of the reconstructed point cloud for (a) cylinder model and (b) packing canister.

Table 3. Relative size estimation for space objects.

Object	Length	Radius	λ^*	λ_{GT}^{**}	Deviation (%)
Cylinder Model	0.62	0.12	2.58	2.5	3.2
Packing Canister	2.09	0.17	6.15	5.90	4.2
Shenzhou-6	0.52	0.21	1.238	1.24	0.1
Tiangong-1	4.57	0.75	3.047	3.10	1.7

* Ratio of the length to the radius; ** Ground truth of the ratio of the length to the radius.

5.4. Reconstruction of Space Object

The results of reconstruction and refinement for Shenzhou-6 and Tiangong-1 are displayed in Figure 12. The 3D point cloud structures of both Shenzhou-6 and Tiangong-1 are well recovered; the outliers and points that are recovered with severe errors in position and direction are filtered after refinement. Highlights, mirror image and darkness might appear in the same area of the object as the view point changes. This might result in incomplete reconstruction due to not enough stable matches, such as the left solar wing of Tiangong-1 shown in Figure 12. The outlines of the recovered objects are shown in Figure 13, and the results of relative size measurement are shown in Table 3. Since Shenzhou-6 has an orbital capsule, a re-entry capsule and a propelling capsule, which have different radii, only the length-to-radius ratio of the orbital capsule is estimated. The deviations are still very small.



Figure 12. Results of reconstruction and refinement for Shenzhou-6 and Tiangong-1. From left to right: the recovered point cloud, the point cloud after being refined and points filtered by refinement.

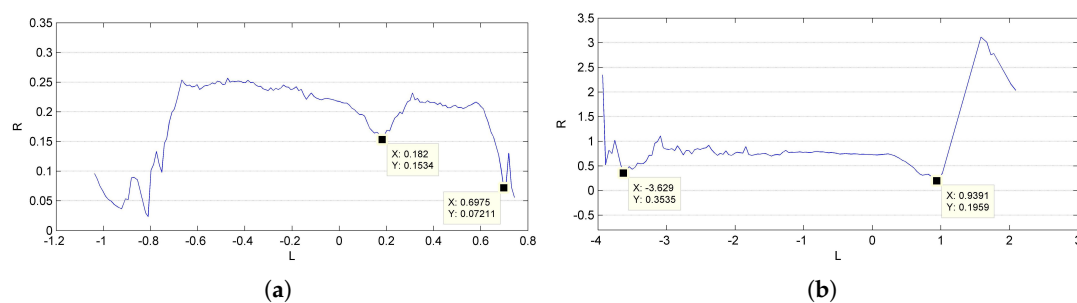


Figure 13. Outline of the reconstructed point cloud for (a) Shenzhou-6 and (b) Tiangong-1.

6. Conclusions

In this paper, we proposed a reconstruction framework for recovering the structure models of space objects using a visible sensor. Given multi-view images of the target object, which could be captured by a visible sensor on space-based surveillance systems, our framework can recover a 3D point cloud model of the target. Such a model can be used to generate full-viewpoint images of the target and is helpful for further estimation and recognition studies. Furthermore, the reconstructed model has an important practical value, as it can be applied to space missions, such as autonomous rendezvous and docking, collision avoidance and on-orbit self-serving. To resolve the incorrect reconstruction, which is caused by the symmetric structure and repeated textures of space objects, we modify the SFM procedure to avoid the unexpected point matches. Meanwhile, a point cloud refinement utilizing the structural prior is introduced to improve the visualization. Experimental results demonstrate that the proposed reconstruction framework can effectively recover a point cloud model of the space object with both a complete coverage and a fine accuracy, and the visualization of the recovered model can be also obviously improved after refinement. In the future, further performance evaluation and improvement will be made for the consideration of the degrading factors, such as noise and blur caused by relative orbital motion. Automatic measurement and recognition of the key components in the reconstructed point cloud, e.g., solar wings, are also topics worth researching.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (Grant No. 61501009, 61371134 and 61071137), the National Key Research and Development Program of China (2016YFB0501300, 2016YFB0501302), the Aerospace Science and Technology Innovation Fund of CASC (China Aerospace Science and Technology Corporation) and the Fundamental Research Funds for the Central Universities.

Author Contributions: H.Z. and Z.J. conceived of and designed the methods and experiments. Q.W. performed the experiments. H.Z. and Q.W. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cefola, P.; Alfriend, K. Sixth US/Russian Space Surveillance Workshop. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; pp. 1677–1693.
2. Jaramillo, C. *Space Security 2010*; Pandora Press: Kitchener, ON, Canada, 2010.
3. Guilmain, B.D. The Midcourse Space Experiment (MSX). In Proceedings of the 1996 IEEE Aerospace Applications Conference, Aspen, CO, USA, 3–9 February 1996; Volume 1, pp. 205–216.
4. Stokes, G.; Vo, C.; Sridharan, R.; Sharma, J. The space-based visible program. In Proceedings of the Space 2000 Conference and Exposition, Long Beach, CA, USA, 19–21 September 2000.
5. Watson, J.; Zondervan, K. The Missile Defense Agency's space tracking and surveillance system. In Proceedings of the SPIE 7106, Sensors, Systems, and Next-Generation Satellites XII, 710617, Cardiff, UK, 15 September 2008.
6. Zhang, H.; Jiang, Z.; Elgammal, A. Vision-Based Pose Estimation for Cooperative Space Objects. *Acta Astronaut.* **2013**, *91*, 115–122.
7. Zhang, X.; Zhang, H.; Wei, Q.; Jiang, Z. Pose Estimation of Space Objects Based on Hybrid Feature Matching of Contour Points. In *Advances in Image and Graphics Technologies, Proceedings of the 11th Chinese Conference, IGTA 2016, Beijing, China, 8–9 July 2016*; Springer: Singapore, 2016; pp. 184–191.
8. Jia, F.; Li, Y.; Sun, H. The Algorithmic Research of Space Object Identification from Multi-viewpoints. *J. Acad. Equip. Command Technol.* **2009**, *6*, 016.
9. Meng, G.; Jiang, Z.; Liu, Z.; Zhang, H.; Zhao, D. Full-Viewpoint 3D Space Object Recognition Based on Kernel Locality Preserving Projections. *Chin. J. Aeronaut.* **2010**, *23*, 563–572.
10. Wang, X. Space Target Recognition Based on Improved Kernel FCM. *Chin. Space Sci. Technol.* **2012**, *32*, 35–42.
11. Ding, H.; Li, X.; Zhao, H. An approach for autonomous space object identification based on normalized AMI and illumination invariant MSA. *Acta Astronaut.* **2013**, *84*, 173–181.
12. Zhang, H.; Jiang, Z. Multi-View Space Object Recognition and Pose Estimation Based on Kernel Regression. *Chin. J. Aeronaut.* **2014**, *27*, 1233–1241.

13. Zhang, H.; Jiang, Z.; Elgammal, A. Satellite Recognition and Pose Estimation Using Homeomorphic Manifold Analysis. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 785–792.
14. Hartly, R.; Zisserman, A. *Multiple View Geometry in Computer Vision Second Edition*; Cambridge University Press: Cambridge, UK, 2004.
15. Tomasi, C.; Kanade, T. Shape and motion from image streams under orthography: A factorization method. *Int. J. Comput. Vis.* **1992**, *9*, 137–154.
16. Bougnoux, S.; Robert, L. Totalcalib: A fast and reliable system for off-line calibration of image sequences. In Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 17–19 June 1997.
17. Shum, H.Y.; Han, M.; Szeliski, R. Interactive construction of 3D models from panoramic mosaics. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 23–25 June 1998; pp. 427–433.
18. Cipolla, R.; Robertson, D.; Boyer, E. PhotoBuilder-3D models of architectural scenes from uncalibrated images. In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, 7–11 June 1999; Volume 1, pp. 25–31.
19. Pollefeys, M.; Gool, L.V. From Images to 3D Models. *Commun. ACM* **2002**, *45*, 50–55.
20. Pollefeys, M.; Van Gool, L.; Vergauwen, M.; Verbiest, F.; Cornelis, K.; Tops, J.; Koch, R. Visual Modeling with a Hand-Held Camera. *Int. J. Comput. Vis.* **2004**, *59*, 207–232.
21. Pollefeys, M.; Nistér, D.; Frahm, J.M.; Akbarzadeh, A.; Mordohai, P.; Clipp, B.; Engels, C.; Gallup, D.; Kim, S.J.; Merrell, P.; et al. Detailed Real-Time Urban 3D Reconstruction from Video. *Int. J. Comput. Vis.* **2008**, *78*, 143–167.
22. Snavely, N.; Seitz, S.M.; Szeliski, R. Photo Tourism: Exploring Photo Collections in 3D. *ACM Trans. Graph.* **2006**, *25*, 835–846.
23. Snavely, N.; Seitz, S.M.; Szeliski, R. Modeling the World from Internet Photo Collections. *Int. J. Comput. Vis.* **2008**, *80*, 189–210.
24. Zhang, H.; Wei, Q.; Zhang, W.; Wu, J.; Jiang, Z. Sequential-image-based space object 3D reconstruction. *J. Beijing Univ. Aeronaut. Astronaut.* **2016**, *42*, 273–279.
25. Ponce, J.; Furukawa, Y. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1362–1376.
26. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
27. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395.
28. Hartley, R.I. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 580–593.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).