

Article

Pose Estimation for Straight Wing Aircraft Based on Consistent Line Clustering and Planes Intersection

Xichao Teng ^{1,*}, Qifeng Yu ¹, Jing Luo ², Xiaohu Zhang ^{1,3} and Gang Wang ¹

¹ College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China; yuqifeng@vip.sina.com (Q.Y.); zhangxiaohu@nudt.edu.cn (X.Z.); wanggang_nudt@163.com (G.W.)

² High-Tech Institute, Qing Zhou 262500, China; luoj11@tsinghua.org.cn

³ School of Aeronautics and Astronautics, Sun Yat-Sen University, Guangzhou 510000, China

* Correspondence: tengari@buaa.edu.cn; Tel.: +86-159-7317-5049

Received: 27 December 2018; Accepted: 11 January 2019; Published: 16 January 2019



Abstract: Aircraft pose estimation is a necessary technology in aerospace applications, and accurate pose parameters are the foundation for many aerospace tasks. In this paper, we propose a novel pose estimation method for straight wing aircraft without relying on 3D models or other datasets, and two widely separated cameras are used to acquire the pose information. Because of the large baseline and long-distance imaging, feature point matching is difficult and inaccurate in this configuration. In our method, line features are extracted to describe the structure of straight wing aircraft in images, and pose estimation is performed based on the common geometry constraints of straight wing aircraft. The spatial and length consistency of the line features is used to exclude irrelevant line segments belonging to the background or other parts of the aircraft, and density-based parallel line clustering is utilized to extract the aircraft's main structure. After identifying the orientation of the fuselage and wings in images, planes intersection is used to estimate the 3D localization and attitude of the aircraft. Experimental results show that our method estimates the aircraft pose accurately and robustly.

Keywords: pose estimation; straight wing aircraft; structure extraction; consistent line clustering; parallel line; planes intersection

1. Introduction

Since the 3D pose parameters of aircraft could provide a lot of valuable information about the aircraft's flight status, effective and accurate pose estimation is a key technique in many aerospace applications, such as autonomous navigation [1], auxiliary landing [2], collision avoidance [3], accident analysis, and testing of a flight control system [4,5]. In recent years, with the development of imaging technology and computer vision, vision-based pose estimation has become a research hotspot, and a lot of methods have been proposed in the literature to estimate the pose of an aircraft using visual sensors. Visual sensors could be successfully applied in aircraft pose estimation since vision-based methods have the advantages of strong anti-interference ability, low cost, and high precision [6].

Vision-based pose estimation methods can be divided into two categories—on-board vision and external vision—depending on the mounting position of the visual sensors. On-board monocular, depth, or stereo cameras can be used in on-board vision methods to estimate the relative pose between the aircraft and a particular target or marker, while external vision methods utilize external cameras to acquire the pose of an aircraft from its 2D projected images.

Among the on-board vision methods, a binocular stereovision model established by Chen et al. [7] used stereo vision and the RANSAC (RANdom SAmple Consensus) algorithm to measure the pose of a non-cooperative target. Li et al. [8] used parallel binocular cameras to estimate the pose of a

non-cooperative target based on stereo matching and 3D restructuring. Zhang et al. [9,10] proposed optimization-based methods to estimate the relative pose using stereo cameras, and the geometric structure of the non-cooperative target was exploited to improve the accuracy. Deng et al. [11] implemented an on-board binocular vision-based system to estimate the pose of Unmanned Aerial Vehicles (UAVs) for autonomous aerial refueling. Zhuang et al. [12] used the line features of the airport and the monocular camera on board to provide pose information for UAV autonomous landing. Benini et al. [13] estimated the pose of a UAV by detecting a marker composed of known circles for autonomous takeoff and landing. For scenes without known landmarks, the structure from motion (SFM) [14–16] method or the simultaneous localization and mapping (SLAM) [17,18] method can be leveraged to estimate the relative pose for aircraft navigation. A sequence of images is processed in these techniques, and a Kalman filter [19,20] is often used to reduce the pose error.

For external vision methods, the aircraft's pose is often estimated using its 2D projected images captured by external imaging devices. Monocular cameras are widely used in external vision systems because the distance between aircraft and cameras is usually large. It is hard to estimate a 3D pose from a single 2D image without prior information such as 3D models of aircraft, synthetic aircraft image datasets, or acquired image sequences. Considering that pose estimation using complete aircraft models viewed from all aspects is storage- and time-consuming, feature extraction and pattern matching methods are proposed to reduce the dimension of pose estimation.

Hmam et al. [21] recognized aircraft based on a geometry-based reasoning system, and a generic model description of the aircraft was used for pose estimation. Wang et al. [22] combined a mathematical morphological algorithm and the Radon transform to extract the aircraft's structure and used the average value of ordinary aircraft as a reference to calculate 3D pose parameters from 2D images. The use of a generic model of aircraft makes these algorithms more efficient and flexible, but this also leads to a reduction in the accuracy and robustness of pose estimation.

Breuers and Reus [23] used a Fourier-descriptor-based algorithm to estimate aircraft pose information. The method computes a Fourier descriptor to characterize the aircraft contour, and the pose information is estimated by comparing this Fourier descriptor to a reference database. Fu et al. [24] estimated the relative pose parameters of aircraft based on a contour model. The method first acquires 2D projections of a 3D model from different views and establishes a database; then, contour matching is employed to derive relative pose parameters. Wang et al. [25] estimated the pose of commercial aircraft in a runway end safety area using geometry structure features. This image-based method obtains aircraft pose information using the central moments of extracted geometry structure features and identifies an aircraft's particular pose by a two-step feature matching strategy. Yuan et al. [26] proposed an aircraft pose recognition method based on locally linear embedding (LLE). In this method, LLE is applied for feature extraction and dimension reduction, and aircraft pose is recognized by propagation neural networks and nearest-neighbor algorithms. Although these methods reduce the complexity of the problem by feature extraction and pattern matching, 3D models of different aircraft are still needed, and a large amount of high-quality training data is necessary for pattern recognition to achieve accurate pose estimation, which reduces the flexibility and efficiency.

While there are a lot of features related to geometric structure to describe the pose information of aircraft, many of them were proposed for swept wing aircraft. Methods for straight wing aircraft structure extraction [27] and pose estimation are seldom addressed, despite the fact that the straight wing and its variants are the most common wing planform for low-speed aircraft [28]. With the rapid development of high-altitude long-endurance (HALE) UAVs, which often adopt a large-aspect-ratio straight wing design in order to increase lift [29,30], pose estimation of straight wing aircraft is of great importance.

In this article, we use a vision system located on the ground to estimate the pose of model-unknown straight wing aircraft. The vision system needs at least two monocular cameras to estimate the 3D pose. There are usually multiple cameras distributed in the flight test site or airport area, which allows our method's requirements to be easily met. Compared to methods which rely on the use of 3D

models and/or classifiers, only two 2D images obtained at the same time and some prior assumptions are explored in our approach to achieve accurate pose estimation for straight wing aircraft. We first identify the orientation of the fuselage and wings in an image pair using consistent line clustering; then, the planes intersection method is used to calculate the 3D pose information of the aircraft.

In the application scenario of this article, the dual-station photoelectric theodolite at a flight test site was used to estimate the absolute pose of an aircraft. The photoelectric theodolite tracks the aircraft, captures a sequence of images, and records the camera pose for every image frame. The image pair captured by the dual-station photoelectric theodolite was used to estimate the pose of the aircraft. To improve the measurement range and accuracy, two photoelectric theodolites with large baseline were selected and distributed on both sides of aircraft trajectory. Because of the large baseline and long-distance measurement, it is very difficult to obtain corresponding invariant features, and self-occlusion at certain angles would make feature matching more unreliable. In order to identify the main structure (fuselage and wings) of the aircraft in image pairs efficiently and robustly, the general geometry features of straight wing aircraft were analyzed.

In our method, line features extracted by the line segment detector (LSD) algorithm are used to describe the structure of the aircraft. The spatial and length consistency of line features is exploited to eliminate the disturbance of the background and unrelated parts of the aircraft, and parallel line segments are grouped into orientation-consistent clusters which represent the structure of the straight wing aircraft. To extract the aircraft's structure accurately and robustly, a density-based clustering method is adopted according to the characteristics of the data. Mean shift and image moment methods are also used to improve the localization accuracy of the aircraft's center in images. After recognizing the main structure of the straight wing aircraft in 2D images, the planes intersection method is used to determine the 3D pose. Our algorithm provides a universal framework to estimate the 3D pose of straight wing aircraft without relying on 3D models, cooperative markers, or other datasets.

The remainder of the paper is organized as follows: Section 2 introduces the coordinate system definition. Our pose estimation algorithm is explained in detail in Section 3. In Section 4, the experimental results of structure extraction and pose estimation are presented to validate our algorithm. Finally, Section 5 concludes this article.

2. Coordinate System Definition

In this section, we define several coordinate systems related to pose estimation. There are three major coordinate systems which are shown in Figure 1.

The world coordinate system (see Figure 1a) helps us track the aircraft and determine its position and attitude. We used the East-North-Up (ENU) coordinate system as the world frame.

The camera coordinate system, shown in Figure 1b, is attached to a camera which tracks the aircraft in the image plane. The origin of the camera frame is located at the optical center of the camera; the x axis is parallel to the horizontal axis of the image plane in the right direction, and the z axis is the optical axis of the camera in the right-handed coordinate system. Two cameras are used in our algorithm and are calibrated with respect to the world coordinate system; their poses are known for each image frame they record.

For the body coordinate system of straight wing aircraft shown in Figure 1c, the origin is located at the center of the aircraft. The x axis points along the fuselage reference line; the y axis is perpendicular to the fuselage plane of symmetry, directed to the right; and the z axis is perpendicular to the plane where the fuselage and wings are located in the right-handed coordinate system. For a straight wing or its variants, the wing edge lines are approximately parallel to each other, while line segments along the fuselage are approximately parallel to the fuselage reference line. Based on these geometry structure features, our pose estimation algorithm extracts the orientation of the fuselage reference line and the wing axis from which we can determine the orientation of the body coordinate axes of straight wing aircraft.

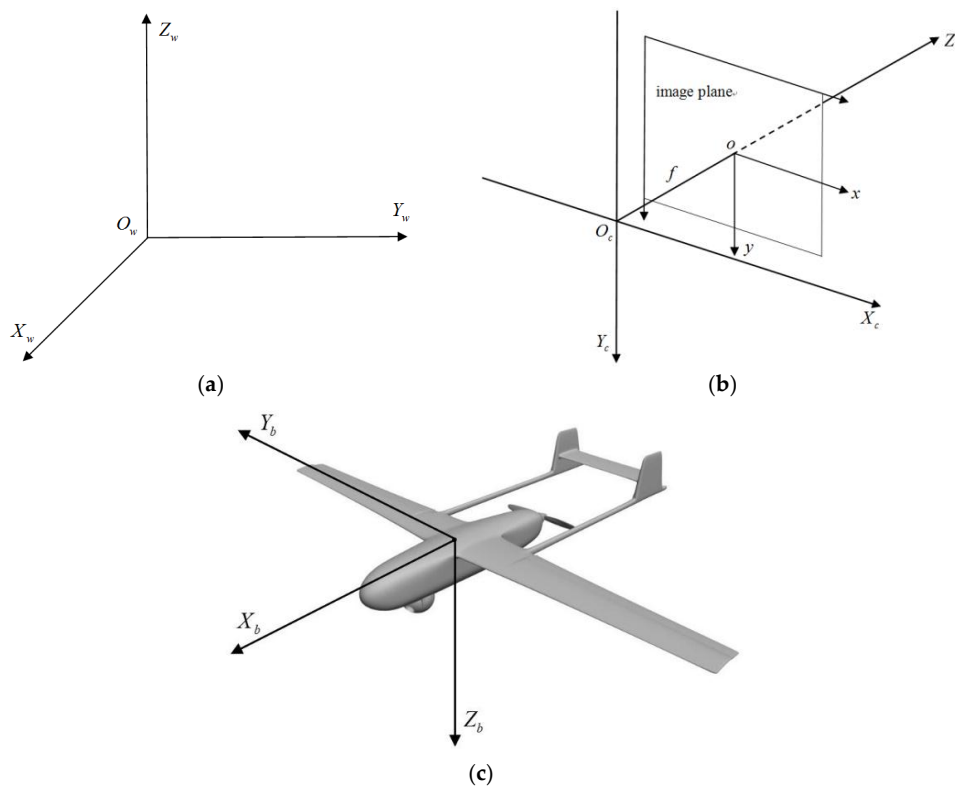


Figure 1. Coordinate systems: (a) World frame; (b) Camera frame; (c) Body frame.

3. Pose Estimation Algorithm

The pose of an aircraft is represented by the transformation using a rotation matrix and translation vector which transform points in the body coordinate frame into points in the world coordinate frame. Our algorithm acquires the 3D pose information of an aircraft by determining the orientation of the body coordinate axes and the position of the body coordinate frame origin with respect to the world coordinate frame.

Our pose estimation algorithm first extracts the orientation of the fuselage and the wings in 2D image pairs, then uses plane–plane intersection to determine the 3D pose of the straight wing aircraft. The 2D pose information acquired by the structure extraction method is used as input to the planes intersection method to acquire the 3D pose of the aircraft. In the process of pose estimation, the initial pose information of the aircraft is needed to avoid ambiguity. In the following sections, the structure extraction and planes intersection methods will be explained in detail.

3.1. Structure Extraction Method

We propose a novel structure extraction method to identify the orientation of the fuselage and wings of straight wing aircraft in a 2D image without needing 3D models or other datasets. Due to the long-range imaging of the aircraft, reliable feature point correspondence is difficult to obtain, especially with ambiguities, extreme poses, or self-occlusions. To obtain the 2D pose information accurately and robustly, we use line features to describe the structure of the aircraft; line features are usually more accurate and robust than feature points in our application scenarios and also adapt to self-occlusions to some extent.

The geometric relations between line features are exploited to recognize the main structure of the aircraft. The most important geometric constraints used in our algorithm are the parallel constraints:

1. Line features distributed along the wing axis are approximately parallel to each other;
2. Line features along the fuselage reference line are approximately parallel to each other.

In addition, line features are concentrated in the area of the aircraft, and the lengths of line features on the main structure (fuselage and wings) of the aircraft are often larger than those on other parts of the aircraft. The main idea of our structure extraction method is to cluster line features based on these geometry constraints to acquire an accurate and robust estimation of the orientation of the aircraft's main structure.

3.1.1. Line Feature Extraction

The state-of-the-art line segment detector (LSD) algorithm is utilized here to extract line features. The LSD algorithm, introduced by Gioi et al. [31], is a linear-time line segment detector giving results to subpixel accuracy, and a comparative study of line extraction methods by Zhang et al. [32] revealed that LSD is an optimal algorithm at different scales, blur degrees, and illumination. We detected 2D line segments using the LSD algorithm in an image to describe the structure of a straight wing aircraft. The result of the line feature extraction is shown in Figure 2a, where red line segments represent the detected line features. We denote the set of detected line segments as S_L .

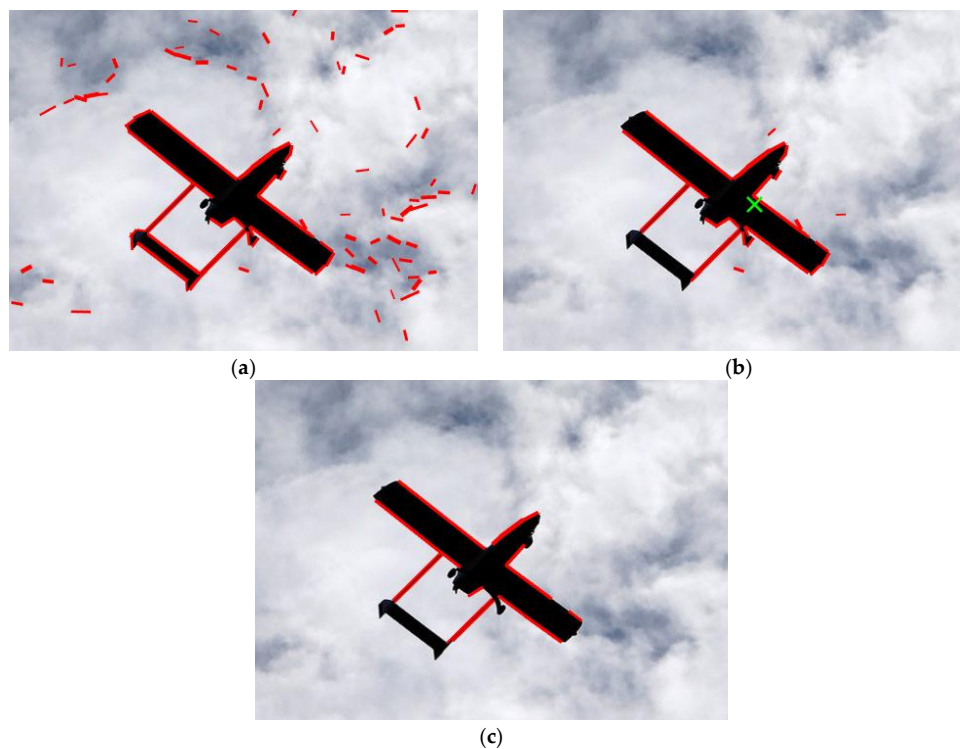


Figure 2. The results of line feature extraction and line clustering based on spatial and length consistency: (a) Line feature extraction; (b) Spatially consistent line clustering; (c) Length-consistent line clustering.

3.1.2. Spatially Consistent Line Clustering

Our algorithm is performed under the condition that the photographic image only contain one aircraft, which is common in actual application scenarios such as flight test, landing, or taking off. As the aircraft is a salient object in the image, detected line segments will be concentrated in the region of the aircraft and close to each other compared to irrelevant line segments, i.e., the density of line segments in the aircraft's region is very high. Based on the location constraint of line segments, we performed spatially consistent line clustering to identify the center of the aircraft and rule out irrelevant line segments caused by the background.

We used the mean shift [33] algorithm to identify the center of the aircraft. Mean shift is a procedure for locating the maxima of a density function given discrete data sampled from that function.

It is useful for detecting the modes of this density, which indicate the spatial consistency of line segments. The set of detected line segments S_L was used as the input of the mean shift algorithm, and a Gaussian kernel \mathbf{K} on the distance was used to determine the weight for re-estimation of the center. An image pixel \mathbf{x} on a line segment which belongs to S_L is represented by (x, y) where x and y are the horizontal and vertical coordinates of the pixel, respectively. The clustering center obtained by the mean shift algorithm is considered the aircraft's center. The kernel function \mathbf{K} and the weighted mean $\mathbf{m}(\mathbf{x})$ of the density can be represented as follows:

$$\begin{aligned} \mathbf{K}(\mathbf{x}_i - \mathbf{x}) &= e^{-c\|\mathbf{x}_i - \mathbf{x}\|^2} \\ \mathbf{m}(\mathbf{x}) &= (\sum_{x_i \in n(\mathbf{x})} K(x_i - x) \cdot x_i) \cdot (\sum_{x_i \in n(\mathbf{x})} K(x_i - x))^{-1} \end{aligned} \quad (1)$$

where c is the weight of the kernel function and $n(\mathbf{x})$ represents the neighborhood of point \mathbf{x} . After determining the center of the aircraft, line segments within a certain distance of the clustering center are considered to belong to the aircraft, and other line segments are removed. Figure 2b shows the result of spatially consistent line clustering. The green cross in Figure 2b represents the cluster centroid of the mean shift, and red line segments indicate the reserved line features which are close to the estimated aircraft's center. As we can see, many line segments which do not belong to the aircraft are rejected by spatially consistent line clustering.

The centroid of the aircraft can also be calculated via image moments, which is given by

$$\tilde{x}_m = \frac{\sum_{i=1}^N x_i}{N}, \quad \tilde{y}_m = \frac{\sum_{i=1}^N y_i}{N}. \quad (2)$$

Here, $(\tilde{x}_m, \tilde{y}_m)$ is the image coordinates of the aircraft's center obtained by the image moment method, and N represents the number of pixels on the line segments. Although the image moment method can identify the centroid of S_L without iteration, it is difficult to obtain the actual center of the aircraft robustly against a cluttered background. Figure 3 shows a comparison of the results of the image moment method and the mean shift algorithm, in which the estimated centroids of the aircraft are indicated by green crosses. The extracted 2D line features (red line segments in Figure 3) were used as the input of both methods. As we can see from Figure 3a, the result of the image moment method deviates from the actual center of the aircraft because of disturbance from the background, while the mean shift algorithm obtained a more accurate aircraft center and is partly resistant to a cluttered background. In the case of a cluttered background, the mean shift algorithm will be used to identify the center of the aircraft, and spatially consistent clustering provides an initial position estimation of the aircraft's center which is then updated in the parallel line clustering.



Figure 3. A comparison of the results of the methods: (a) Image moment method; (b) Mean shift algorithm.

3.1.3. Length-Consistent Line Clustering

Although many noisy line segments are removed by spatially consistent line clustering, there are still some irrelevant line segments, as shown in Figure 2b. In this section, we use a length consistency criterion to further rule out irrelevant line segments. As the aircraft's main structure (fuselage and wings) is usually larger than other parts such as tail or nose, line segments shorter than a certain threshold can be removed from the set of line segments. As the length of a line segment decreases, the uncertainty of its direction increases, i.e., a small position error of the endpoint causes greater direction error for shorter line segments. Excluding shorter line segments would improve the accuracy of 2D pose estimation in the following parallel line clustering.

The result of length-consistent line clustering is shown in Figure 2c. Compared to Figure 2b, the irrelevant line segments are excluded further, which is of benefit for the following clustering.

3.1.4. Parallel Line Clustering

Parallel line clustering is the key step in the structure extraction algorithm and can acquire the directions of the fuselage and wings without relying on 3D models, other datasets, or cooperative markers. Line segments with similar directions are divided into one orientation-consistent line cluster. In the parallel line clustering process, the direction of a line segment is represented by the angle θ_i between the straight line that it belongs to and the horizontal axis of the image plane. The set of directions of line segments $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$ is used as input to the parallel line clustering. The directions of the fuselage and the wings are denoted θ_f and θ_w respectively.

Weak perspective projection (scaled orthographic projection) is employed in parallel line clustering. As the size of the aircraft is small compared with its distance to the optical center along the optical axis, weak perspective approximation is valid. If line segments on the aircraft are parallel to each other in 3D space (the world frame), then this geometry feature of the corresponding line segments projected into the image plane remains unchanged under weak perspective projection.

For straight wing aircraft, line segments distributed along the wings are roughly parallel to each other, and angle values of these line segments are tightly concentrated around θ_w (small standard deviation), while angle values of line segments along the fuselage are concentrated around θ_f . The directions of the fuselage and the wings can be extracted by clustering the high-density regions of Θ . According to the orientation feature of the line segments, a density-based clustering algorithm, density-based spatial clustering of application with noise (DBSCAN) [34], was used to group the parallel line segments into one orientation-consistent cluster containing the orientation information of the fuselage or the wings.

The data points used in DBSCAN clustering are the directions of line segments θ_i . There are two parameters required to be specified in the DBSCAN algorithm, both of which are used to measure the density of data points. The first parameter is a distance threshold ϵ within which two data points close to each other will be grouped into one cluster. The distance threshold is the absolute difference between angle values in our algorithm. The second parameter is the minimum number of data points $minPts$ needed to form an orientation-consistent cluster. Based on these two parameters, the data points are classified into three types, as shown in Figure 4:

- Core points: If a data point's ϵ neighborhood contains at least $minPts$ points, it is a core point (red points in Figure 4);
- Border points: If a data point's ϵ neighborhood contains fewer than $minPts$ points, but it is reachable from a certain core point (as indicated by one-way arrows in Figure 4), it is a border point (yellow points in Figure 4, the edge of a cluster);
- Noise points: If a data point is neither a core point nor a border point, it is a noise point (blue points in Figure 4).

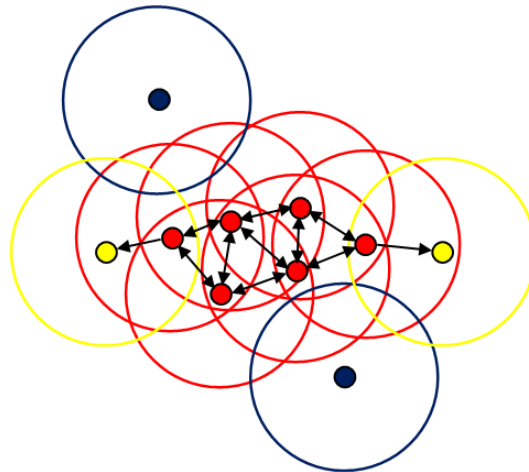


Figure 4. The explanation of the density-based spatial clustering of application with noise (DBSCAN) algorithm: the red points represent the core points, the yellow points represent the border points, and the blue points represent the noise points. The radius of the circle represents the distance threshold.

The steps of the DBSCAN algorithm used for parallel line clustering are briefly described as follows:

1. For every data point θ_i , search points in its ϵ neighborhood and use $minPts$ to determine the core points in the set Θ .
2. Ignore all non-core points and group core points into parallel line clusters based on the connected components on the neighborhood graph (as indicated by two-way arrows in Figure 4).
3. For every non-core point, if it is in the ϵ neighborhood of a cluster, it is the border point of the cluster; otherwise, it is a noise point.

In contrast to traditional clustering methods such as k -means++ [35], the DBSCAN algorithm does not need to specify the number of clusters in advance and is robust to outliers. It forms clusters based solely on the spatial density of the data. The two orientation-consistent clusters with minimum interclass variance represent the structure of the fuselage and the wings which contain the orientation information of the aircraft in the image. The directions of the fuselage and the wings are obtained by extracting the centers of the parallel line clusters.

In our method, the directions of the fuselage and the wings are distinguished based on an initial pose constraint. The approximate orientation of the aircraft needs to be specified in the initial frame of the image sequence to avoid ambiguity and to help identify the actual pose of the aircraft. With this condition, the directions of the fuselage and the wings can be distinguished in the initial frame, and the orientation information of the current frame will be used in the next frame. In the application scenarios of our algorithm, such as take-off, landing, or flight testing, this condition is easily met. In practice, the pitch angle (or the yaw angle) and the roll angle of the aircraft are provided, or the approximate positions of the nose and one wing tip are marked in one image of the initial image pair.

After obtaining the orientation-consistent clusters, irrelevant line segments are removed from the set S_L , and the position of the aircraft's center is then re-estimated from the set S_L based on the image moment method. Since only line segments on the main structure of the aircraft are left, it is possible to identify the center of the aircraft with higher precision.

The results of parallel line clustering are shown in Figure 5. As shown in Figure 5a, the red straight lines indicate the directions of the fuselage and the wings, and the green cross indicates the estimated centroid of the aircraft. The directions of the fuselage and the wings were correctly extracted by parallel line clustering. However, in Figure 5b, there is only one cluster with enough parallel line segments for this extreme pose. In this case, the direction of only the fuselage or the wings can be

acquired by parallel line clustering, and the unknown direction needed for pose estimation is replaced by the corresponding orientation information of the previous frame.

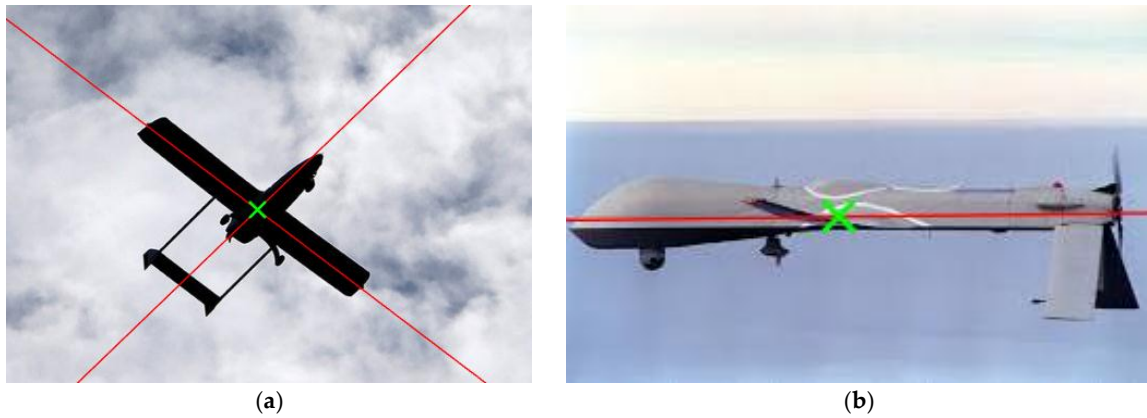


Figure 5. The results of parallel line clustering. (a) The directions of the fuselage and the wings are extracted correctly; (b) The direction of the wings cannot be extracted for this extreme pose.

3.2. Planes Intersection Method

After the structure extraction method determines the pixel coordinates of the aircraft's center and the directions of the fuselage and the wings in an image pair, the planes intersection method is used to estimate the 3D pose of the aircraft.

Two cameras were used in the intersection measurement and are indicated by their projection matrices \mathbf{P}_1 and \mathbf{P}_2 . The camera projection matrices are of the form

$$\mathbf{P}_1 = \mathbf{K}_1[\mathbf{R}_1|\mathbf{t}_1] \quad \mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2] \quad (3)$$

where \mathbf{K}_i ($i = 1, 2$) is the camera intrinsic matrix of the camera, and \mathbf{R}_i and \mathbf{t}_i represent the rotation and translation, respectively, of the corresponding camera with respect to the world frame. We assume that the cameras are calibrated with respect to the world frame and that the \mathbf{P}_i are known.

The camera model is represented as

$$z\mathbf{x} = \mathbf{P}_i\mathbf{X} \quad (4)$$

where $\mathbf{X} = (X, Y, Z, 1)^T$ is the world coordinates and $\mathbf{x} = (u, v, 1)^T$ is the image coordinates of \mathbf{X} . As weak perspective projection is employed, z is a positive constant.

The image pair captured by the two cameras at the same time is denoted $\langle I_1, I_2 \rangle$. The center of the aircraft obtained by the structure extraction method in the image I_i ($i = 1, 2$) is represented as $AC_i = (x_i, y_i)$ where x_i and y_i are the horizontal and vertical coordinates of the image, and the directions of the fuselage and the wings are represented as θ_i^f and θ_i^w , respectively.

Figure 6 explains the geometric constraint of the planes intersection method. As shown in Figure 6, the two cameras are indicated by their optical centers \mathbf{C}_1 and \mathbf{C}_2 and by image planes. The 3D line in the world coordinate system is represented as \mathbf{L} , which is the line of intersection of the two planes π_1 and π_2 ; l_i ($i = 1, 2$) is the projected line of \mathbf{L} in the image plane; and the plane π_i is determined by the line \mathbf{L} and the optical center \mathbf{C}_i . Let the normalized vector \mathbf{V} of \mathbf{L} represent the direction of the fuselage or the wings; the planes intersection method estimates the 3D attitude of the aircraft by obtaining the solution of \mathbf{V} .

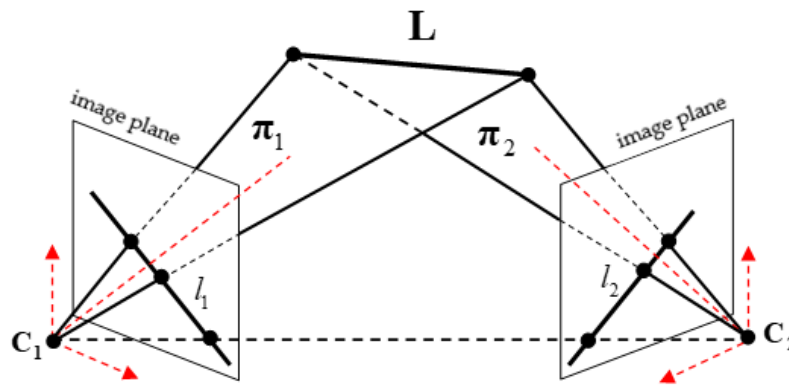


Figure 6. Geometric constraint of plane–plane intersection.

The projected line l_i in the image plane is identified by AC_i and θ_i^f (or θ_i^w); an analytical expression of l_i is

$$a_i u + b_i v + c_i = 0. \quad (5)$$

Equation (5) can be represented in vector form as the following:

$$\begin{bmatrix} a_i & b_i & c_i \end{bmatrix} \mathbf{x} = 0. \quad (6)$$

By substituting Equation (6) into Equation (4) for each camera, we obtain

$$z \begin{bmatrix} a_i & b_i & c_i \end{bmatrix} \mathbf{x} = \begin{bmatrix} a_i & b_i & c_i \end{bmatrix} \mathbf{P}_i \mathbf{X} = 0, \quad (7)$$

and the plane π_i can be expressed as

$$\begin{aligned} & \begin{bmatrix} A_i & B_i & C_i & D_i \end{bmatrix} \mathbf{X} = 0 \\ \vec{\mathbf{n}}_i &= \begin{bmatrix} A_i & B_i & C_i \end{bmatrix} \end{aligned} \quad (8)$$

where $\vec{\mathbf{n}}_i$ is the normalized vector of the plane π_i . Note that Equations (7) and (8) have the same form, and $\begin{bmatrix} a_i & b_i & c_i \end{bmatrix} \mathbf{P}_i$ is already known; the normalized vector $\vec{\mathbf{n}}_i$ is derived from Equations (7) and (8). After we obtain the normalized vectors $\vec{\mathbf{n}}_1$ and $\vec{\mathbf{n}}_2$, the normalized vector \mathbf{V} which contains the orientation information of the aircraft is solved as follows:

$$\mathbf{V} = \vec{\mathbf{n}}_1 \times \vec{\mathbf{n}}_2. \quad (9)$$

As the 3D line \mathbf{L} can be parametrized in the world coordinate frame by the two planes π_1 and π_2 as a 2×4 matrix, let \mathbf{L}_f be the 3D line parallel to the fuselage reference line and \mathbf{L}_w be the 3D line parallel to the wing edge lines. The point of intersection of \mathbf{L}_f and \mathbf{L}_w is the center of the aircraft. By calculating the respective normalized vectors of \mathbf{L}_f and \mathbf{L}_w using Equation (9), we can obtain the 3D attitude of the straight wing aircraft. The rotation matrix is calculated by singular value decomposition, and the initial pose constraint is used to avoid reflective ambiguity. In order to obtain the world coordinates of the point of intersection of \mathbf{L}_f and \mathbf{L}_w , which determine the 3D position of the aircraft, overdetermined equations are established as follows.

$$\begin{aligned} \mathbf{A}\mathbf{X} &= 0 \\ \mathbf{A} &= \begin{bmatrix} \mathbf{L}_f \\ \mathbf{L}_w \end{bmatrix} \end{aligned} \quad (10)$$

Here, \mathbf{A} is a 4×4 matrix and \mathbf{X} represents the point of intersection of the two lines (the translation vector). The overdetermined equations $\mathbf{AX} = 0$ can be solved by singular value decomposition, and the solution is the singular vector corresponding to the smallest singular value of \mathbf{A} . Before solving the overdetermined equations, an optimal estimator for the center point based on the epipolar constraint can be used to reduce the geometric error [36].

The 3D attitude of the aircraft is determined by the normalized vectors of \mathbf{L}_f and \mathbf{L}_w , and the 3D position of the aircraft is determined by the point of intersection \mathbf{X} of the two lines. As we assume that \mathbf{L}_f and \mathbf{L}_w are coplanar in our pose estimation algorithm, the ambiguity will occur during the process of pose estimation, and the initial pose constraint will be used to determine the unique solution. Based on the results of the structure extraction method, the planes intersection method can acquire the 3D pose of the straight wing aircraft. Moreover, our pose estimation algorithm can easily be extended to multiple camera views.

3.3. Algorithm Summary

In this section, we summarize the whole pose estimation algorithm as is shown in Algorithm 1.

Algorithm 1: Pose estimation based on consistent line clustering and planes intersection

Input:	The image pair $\langle I_1, I_2 \rangle$, the two camera matrices $\mathbf{P}_1, \mathbf{P}_2$, and the initial pose constraint.
Output:	The 3D position and 3D attitude of the straight wing aircraft.
Step 1	Extract line features in image pairs using the LSD algorithm;
Step 2	Locate the center of the aircraft in the 2D images and cluster spatially consistent line segments;
Step 3	Rule out line segments shorter than a certain threshold;
Step 4	Classify line segments into orientation-consistent clusters, extract the directions of the fuselage and the wings in the image pair, and re-estimate the center of the aircraft;
Step 5	Calculate the 3D attitude and 3D location using the plane–plane intersection method.

The flowchart of the algorithm is shown in Figure 7.

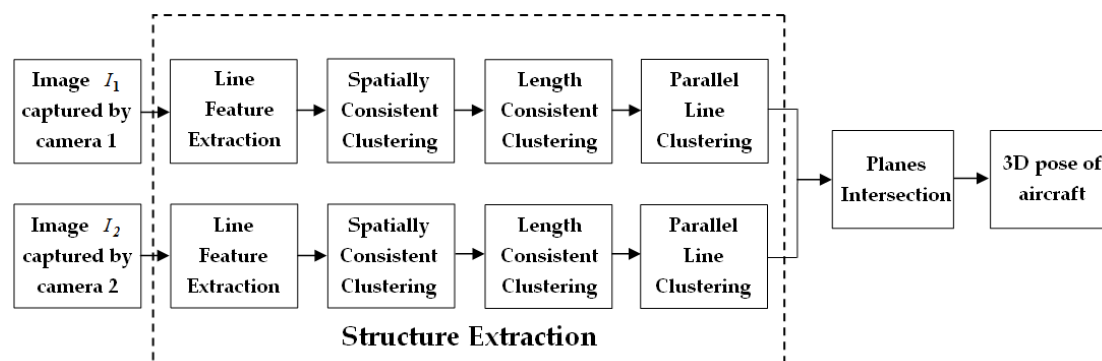


Figure 7. The flowchart of our pose estimation algorithm.

4. Experiments and Results

Experiments were performed to validate the effectiveness and accuracy of the proposed structure extraction and pose estimation methods. Real images of different straight wing aircraft downloaded from the Internet were used to demonstrate the effectiveness and universality of the structure extraction method, and simulated images of straight wing aircraft were exploited to evaluate the accuracy of our pose estimation algorithm. Our method was implemented using MATLAB on a laptop equipped with an Intel Core i7 CPU with a 2.80 GHz processor and 8.00 GB of RAM.

4.1. Experimental Results of Structure Extraction

The qualitative evaluation of our structure extraction method was performed using real images downloaded from the Internet. A total of 60 images of different sizes were downloaded and used in the experiment. Each image contains one aircraft whose planform is the straight wing or its variant, and the structure extraction method was used to identify the orientation of the aircraft's main structure in a single 2D image. Among these images, some are challenging for structure extraction since they contain a cluttered background, other objects, random noise, or perspective effects.

In the experiment, the directions of the fuselage and the wings in 51 of the 60 images were correctly identified. Figure 8 shows some of the results of structure extraction. As we can see, our structure extraction algorithm can be applied flexibly to different types of straight wing aircraft without needing 3D models of aircraft or other datasets, and it can also deal with different aircraft poses effectively and robustly extract the main structure under self-occlusion or a cluttered background. Moreover, the parallel assumption does not need to hold strictly. Even if perspective effects exist or the line segments are not strictly parallel to each other, our algorithm can still recognize the main structure of the aircraft.



Figure 8. Results of the structure extraction method.

While the algorithm achieved good results in most downloaded images, Figure 9 shows some cases in which our structure extraction method obtained incorrect results. There are two main reasons for these incorrect results:

1. The structure of the aircraft (fuselage or wings) does not satisfy the assumption of parallel line clustering, i.e., the line segments distributed along this structure are not parallel to each other in the image (see row 1, Figure 9).
2. Some parts of the aircraft (tail or external mounts) or the background affect the consistent line clustering (see row 2, Figure 9).

Changing weather or light conditions may also affect the success rate of our algorithm. When the weather condition or brightness/darkness level changes, the edges of the aircraft's main structure may be blurred during image acquisition, and unreliable line features will be detected. Changing weather or light conditions may affect the accuracy and robustness of the line feature detection, which in turn disturbs the consistent line clustering results and reduces the accuracy and success rate of our

algorithm. In our method, the LSD algorithm used to detect line features can adapt to optical blur and illumination changes to some extent.



Figure 9. Some incorrect results from our structure extraction algorithm.

The situations shown in Figure 9 are uncommon in our application scenarios, and despite the fact that our algorithm is mainly for estimating the pose of a straight wing aircraft at long distance, the experimental results show that the proposed algorithm is able to recognize the aircraft's main structure robustly even at close range.

4.2. Experimental Results of Pose Estimation

Simulated image pairs were used to test our pose estimation algorithm. Two models were used in our experiment to simulate straight wing aircraft, as shown in Figure 10. These two models were created using Autodesk 3ds Max [37], which is a professional 3D computer graphics program for making 3D animations, models, and images. Model 1 (see Figure 10a) represents a general commercial UAV with standard straight wings while Model 2 (see Figure 10b) is a full-size simulation of the MQ-9 unmanned aircraft which has straight tapered wings (a variant of the standard straight wing). The size of Model 1 is 3.4 m × 5.0 m × 0.7 m (length, width, height), and the size of Model 2 is 10.4 m × 24.8 m × 3.1 m (length, width, height).

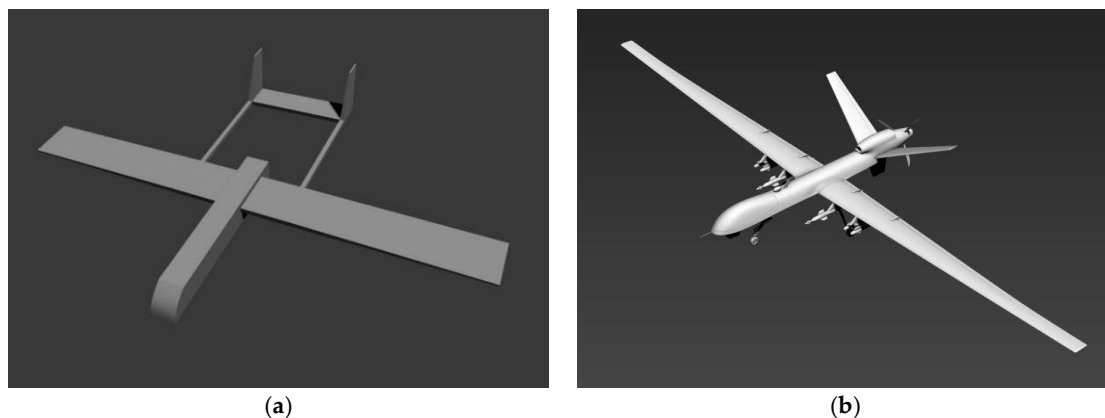


Figure 10. Two aircraft models: (a) Model 1; (b) Model 2.

Two cameras in 3ds Max were used to simulate the dual-station photoelectric theodolite at the flight test site. The internal parameters and spatial layouts of the cameras for aircraft pose estimation are shown in Table 1. As we can see from Table 1, flight simulation scenarios were established for Model 1 (Scene 1, see Table 1) and Model 2 (Scene 1, see Table 1).

Table 1. The internal parameters and spatial layouts of the cameras in Scene 1 and Scene 2.

	Camera	Focal Length	Field of View	Image Resolution	Location (x,y,z)
Scene 1	1	70 mm	28.842° × 21.832°	1280 × 960	(−15 m, −25 m, 0)
	2	75 mm	26.991° × 20.408°	1280 × 960	(−20 m, 30 m, 0)
Scene 2	1	300 mm	6.867° × 5.153°	1280 × 960	(350 m, 550 m, 30 m)
	2	275 mm	7.49° × 5.621°	1280 × 960	(170 m, −390 m, 0)

In our simulation experiments, cameras with different internal parameters and spatial layouts were used to test the performance of our algorithm, and the two cameras in the scene were located on both sides of the aircraft trajectory. The location coordinates of the cameras were in the East-North-Up (ENU) coordinate system. In Scene 1, the baseline between the two cameras was 55.23 m, while the two cameras in Scene 2 had a baseline of 957.55 m. The image pairs were generated by the two cameras in the scenes, and it is very difficult to obtain reliable feature correspondences in these wide-baseline images.

In order to test the performance of our pose estimation algorithm on different poses in simulation image pairs, we rotated Model 1 around the x , y , and z axes to simulate changes in the roll angle γ , pitch angle ψ , and yaw angle φ , respectively. Table 2 shows the selected rotation angles ($\theta_x, \theta_y, \theta_z$) of Model 1, where θ_x represents the roll angle, θ_y represents the pitch angle, and θ_z represents the yaw angle. As detailed in Table 2, 13 image pairs were generated for Model 1, and the selected angle range was reasonable considering actual flight situations. The translation vector of Model 1 in Scene 1 was $\mathbf{T}_{true} = (0, 0, 20 \text{ m})$.

Table 2. The selected rotation angles of Model 1 in Scene 1.

	1	2	3	4	5	6	7	8	9	10	11	12	13
θ_x	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°	−15°	15°
θ_y	0°	0°	0°	0°	0°	0°	0°	−30°	30°	−15°	15°	0°	0°
θ_z	0°	−30°	30°	−60°	60°	−90°	90°	0°	0°	0°	0°	0°	0°

For Model 2 in Scene 2, an aircraft trajectory was designed to simulate the flight. During the flight simulation, the pitch angle of Model 2 varied from $−15^\circ$ to 15° , the roll angle varied from $−10^\circ$ to 10° , and the translation vector was $\mathbf{T}_{true} = (x, 0, 200 \text{ m})$, where x ranged from 0 to 600 m. The simulated flight path was rendered into 13 image pairs in steps of 50 m in Scene 2, and the rotation angles of each step are shown in Table 3.

Table 3. The rotation angles of Model 2 in Scene 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13
θ_x	0°	0°	0°	−10°	−5°	−5°	10°	0°	5°	5°	0°	0°	0°
θ_y	0°	0°	0°	0°	15°	10°	−5°	5°	−5°	0°	−15°	−10°	0°
θ_z	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°	0°

We used the 3ds Max rendering engine to generate the simulated image pairs of these two models; these are shown in Figure 11a,b. In Figure 11a, the top row and the bottom row represent the simulated images of Model 1 captured by Camera 1 and Camera 2, respectively, in Scene 1, and every column represents an image pair captured at the same time. In Figure 11b, the top row and the bottom row represent the simulated images of Model 2 captured by Camera 1 and Camera 2, respectively,

in Scene 2, and every column represents an image pair captured at the same time. The rotation angles $(\theta_x, \theta_y, \theta_z)$ of the aircraft in each shot are also displayed in Figure 11.

In order to make the simulation scenes more realistic, a sky background with clouds and different types of natural light was also simulated (see Figure 11). As shown in Figure 11, the wide-baseline image pairs contain aircraft with different scales, poses, and self-occlusion, and optical blur exists due to long-range imaging. Under these challenging circumstances, a robust algorithm is needed to obtain accurate pose information from a single image pair.

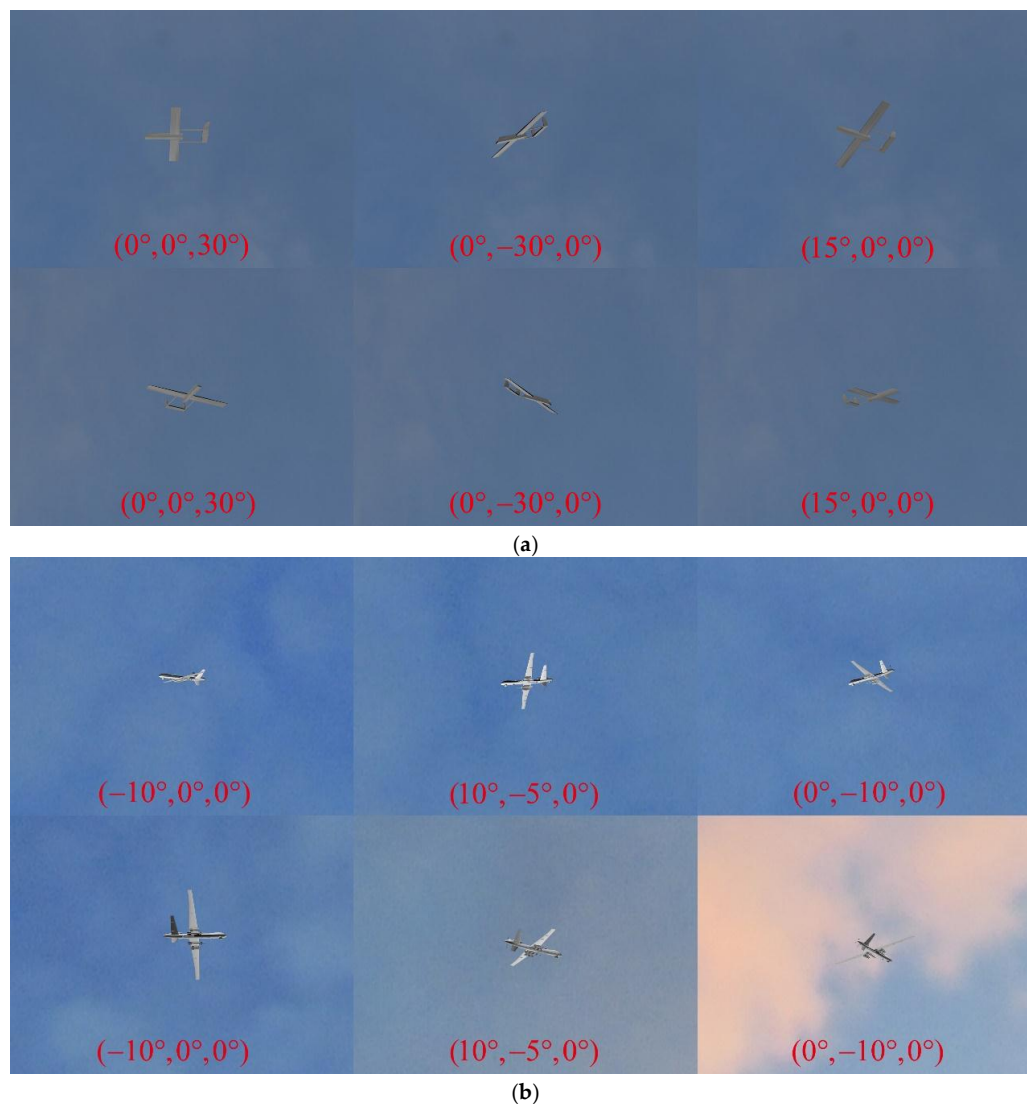


Figure 11. Examples of simulated image pairs generated by the 3ds Max rendering engine: (a) Image pairs of Model 1; (b) Image pairs of Model 2.

Figure 12 presents the results of our structure extraction method on the simulated image pairs shown in Figure 11. The directions of the fuselage and the wings are indicated by the red lines in Figure 12. As we can see, the main structure of the aircraft was correctly extracted by our structure extraction method, and the results further validate the performance of our method. The 3D pose of the aircraft can be obtained effectively only when the 2D pose information in the image pair is extracted robustly and accurately.

We compared our pose estimation algorithm with Li's method [8] and pose estimation errors were used to evaluate the algorithms. In Li's method, the 3D pose of a non-cooperative target is estimated by a stereo camera based on a triangulation method, and the feature points obtained by the line feature

extraction are used for stereo matching and 3D reconstruction. The triangulation method is typically applied to estimate 3D position in computer vision, and the pose estimation pipeline of Li's method is also widely used, so it was selected for comparison to validate our proposed method.

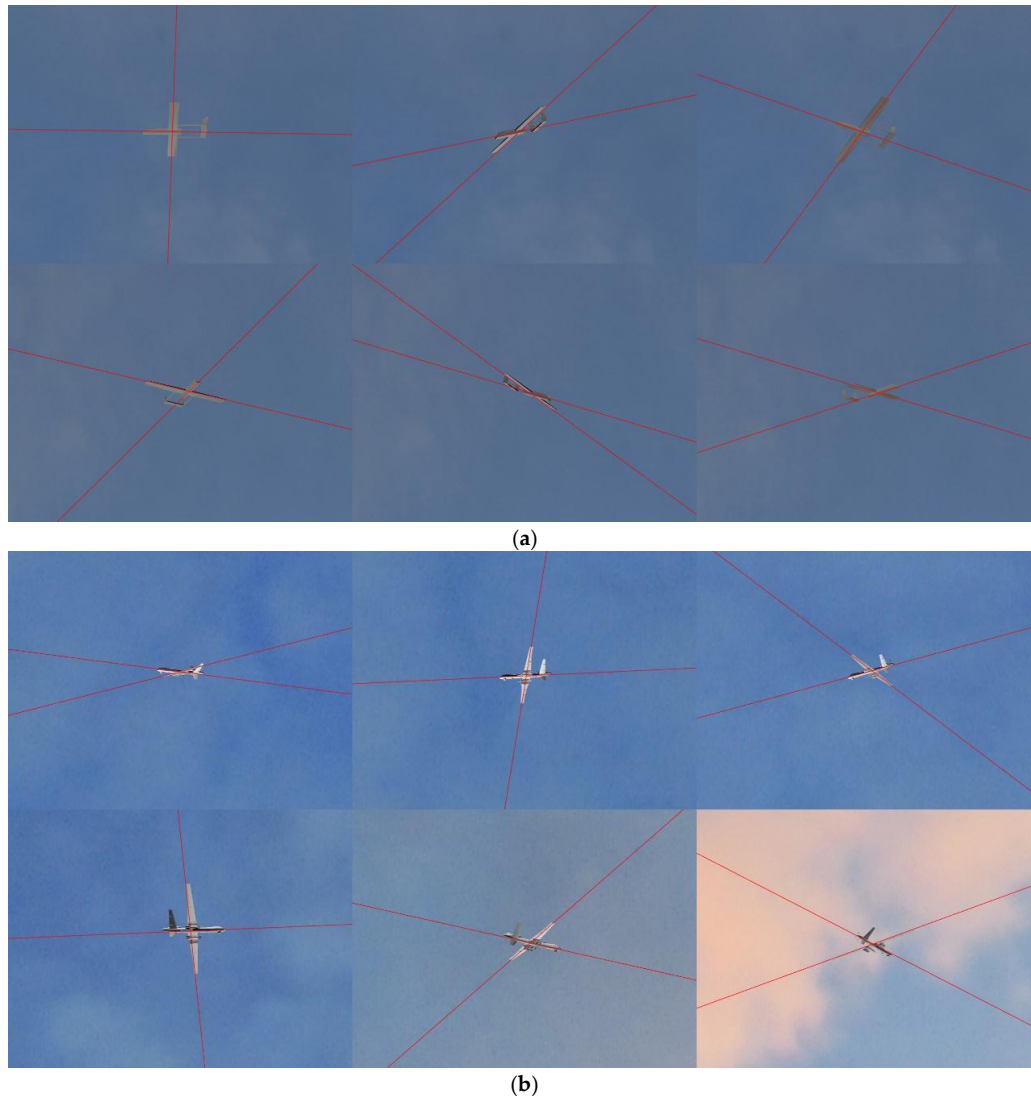


Figure 12. Structure extraction results on the simulated images: (a) Results on Model 1's simulated image pairs; (b) Results on Model 2's simulated image pairs.

For the ground truth pose of the aircraft (\mathbf{R}_{true} and \mathbf{T}_{true}) and corresponding estimated pose ($\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$), the rotation error is calculated by $error_{rot} = \|\hat{\theta} - \theta_{true}\|$ where $\hat{\theta}$ and θ_{true} are the Euler angles of $\hat{\mathbf{R}}$ and \mathbf{R}_{true} , respectively, and the translation error is calculated by $error_{trans} = \|\hat{\mathbf{T}} - \mathbf{T}_{true}\|$.

Since Li's method can hardly obtain reliable feature matching results across these wide-baseline views in our experiments, we manually removed mismatched features, selected correct matches in the image pairs, and confirmed that there were enough corresponding feature points for pose estimation. The 3D models are also used in Li's method to obtain the absolute pose of the aircraft, while our algorithm is model free and acquires the 3D pose information automatically.

Figure 13 shows the pose estimation errors of our algorithm and Li's method for the simulated images of Model 1. In Figure 13a, the rotation errors are presented, and the translation errors are shown in Figure 13b. As we can see from Figure 13, the translation accuracy of our method is similar to that of Li's method, and our pose estimation method performs consistently better than the compared

method in the estimation of the rotation angle. The rotation angle errors of our method are within 1° , the average rotation error is 0.47° , and the average translation error is 177.91 mm.

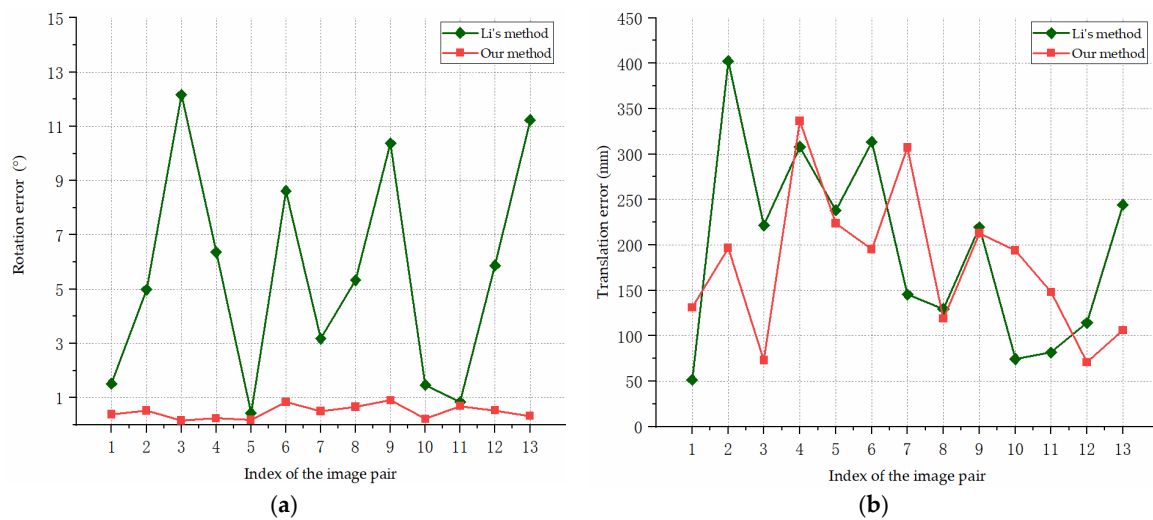


Figure 13. Pose estimation errors for the simulated images of Model 1: (a) Rotation errors; (b) Translation errors.

Figure 14 shows the pose estimation errors of our algorithm and the compared method for the simulated images of Model 2. In Figure 14a, the rotation errors are presented, and the translation errors are shown in Figure 14b. Model 2 is more complex than Model 1 and there is a greater imaging distance in Scene 2, while our algorithm still achieves accurate and stable pose estimation results compared to the results for Model 1. In Figure 14, the proposed method outperforms the compared method in the estimation of the rotation angle and translation vector, which is due to the accuracy and robustness of our structure extraction and planes intersection methods. The large fluctuations in the result curves indicate that the triangulation process used in Li's method is sensitive to various errors. The triangulation method uses the intersection of two lines to estimate the 3D position; with the measure distance increasing, the uncertainty increases, making the results more sensitive to noise. The average rotation error of our method is 1.21° , and the average translation error is 336.49 mm. The experimental results indicate that our method can extract the structure and estimate the pose accurately.

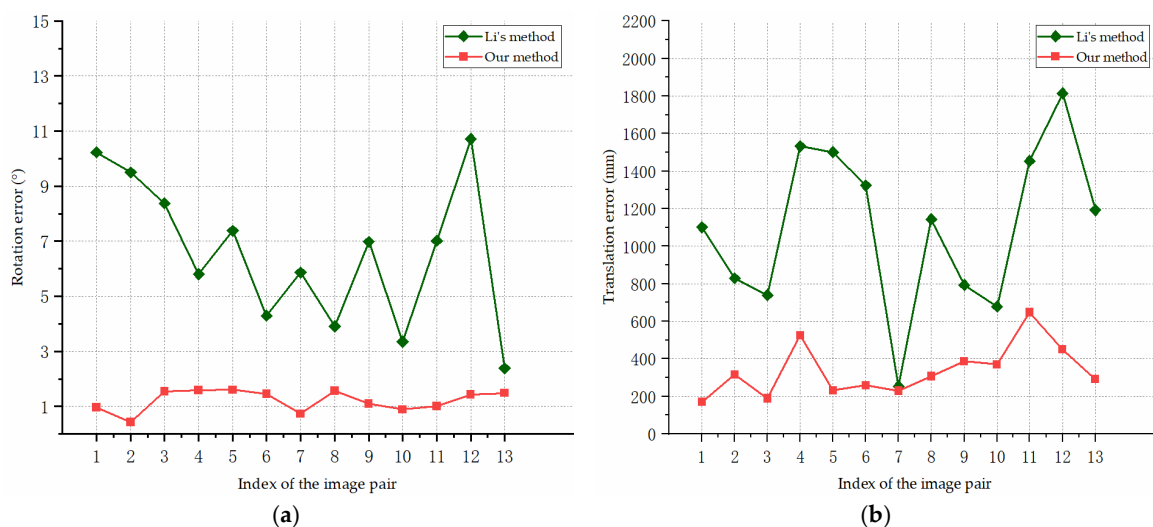


Figure 14. Pose estimation errors for the simulated images of Model 2: (a) Rotation errors; (b) Translation errors.

In addition, our method is also efficient. We ran our method 1000 times and recorded the execution time. The average execution time was 30.74 ms (including the structure extraction and planes intersection methods), which means that our algorithm can estimate the 3D pose efficiently.

The simulation experiment results show that our algorithm estimates the pose of the straight wing aircraft more accurately and robustly than does the compared method. Meanwhile, our method is efficient and flexible and can be applied to different types of straight wing aircraft.

5. Conclusions

An accurate and robust pose estimation method for straight wing aircraft was proposed in this paper. The geometry structure features of straight wing aircraft were utilized for structure extraction and the pose information was acquired by the planes intersection method. Our method establishes a universal framework for pose estimation of straight wing aircraft without relying on 3D models or other datasets, unlike other existing methods, and can be extended to other targets with similar geometric constraints. For an aircraft without similar geometric constraints to straight wing aircraft, our proposed method is unable to extract its main structure robustly and accurately. In the case of a swept wing aircraft, only the fuselage contains enough parallel lines can be detected effectively, while the wings cannot be extracted accurately. Extending our algorithm to aircraft with different wing planforms will be the focus of our future research.

Our method can also provide initial pose information for algorithms with higher precision efficiently. For an image sequence captured during flight, our future work will also focus on using an extended Kalman filter or particle filter to improve the accuracy of our algorithm.

Author Contributions: Conceptualization, X.T.; Methodology, X.T., Q.Y. and J.L.; Software, X.T.; Validation, X.Z.; Investigation, G.W.; Resources, X.Z.; Writing—Original Draft Preparation, X.T.; Writing—Review and Editing, X.T., Q.Y. and J.L.; Visualization, X.T.

Funding: This research was funded by Scientific Research Program of National University of Defense Technology (number: ZK16-03-27), Development and application of vision based micron-sized high speed quality tester (Grant number: 2013YQ140517), and National Natural Science Foundation of China (Grant number: 11727804).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kok, M.; Hol, J.D.; Schön, T.B. Using inertial sensors for position and orientation estimation. *Found. Trends Signal Process.* **2018**, *11*, 1–153. [[CrossRef](#)]
2. Santos, N.P.; Melício, F.; Lobo, V.; Bernardino, A. A ground-based vision system for UAV pose estimation. *Int. J. Robot. Mechatron.* **2015**, *1*, 138–144. [[CrossRef](#)]
3. Langelaan, J.W. State Estimation for Autonomous Flight in Cluttered Environments. *J. Guid. Control Dyn.* **2007**, *30*, 1414–1426. [[CrossRef](#)]
4. Ward, D.G.; Monaco, J.F.; Bodson, M. Development and flight testing of a parameter identification algorithm for reconfigurable control. *J. Guid. Control Dyn.* **1998**, *21*, 948–956. [[CrossRef](#)]
5. Proud, A.W. Close formation flight control. *J. Guid. Control Dyn.* **1999**, *24*, 246–254. [[CrossRef](#)]
6. Yang, Z.; Li, C. Review on vision-based pose estimation of UAV based on landmark. In Proceedings of the IEEE International Conference on Frontiers of Sensors Technologies (ICFST), Shenzhen, China, 14–16 April 2017.
7. Chen, L.; Guo, B.; Sun, W. Relative pose measurement algorithm of non-cooperative target based on stereo vision and RANSAC. *Int. J. Soft Comput. Softw. Eng.* **2012**, *2*, 26–35. [[CrossRef](#)]
8. Li, R.; Zhou, Y.; Chen, F.; Chen, Y. Parallel vision-based pose estimation for non-cooperative spacecraft. *Adv. Mech. Eng.* **2015**, *7*. [[CrossRef](#)]
9. Zhang, L.; Zhu, F.; Hao, Y.; Pan, W. Optimization-based non-cooperative spacecraft pose estimation using stereo cameras during proximity operations. *Appl. Opt.* **2017**, *56*, 4522–4531. [[CrossRef](#)] [[PubMed](#)]

10. Zhang, L.; Zhu, F.; Hao, Y.; Pan, W. Rectangular-structure-based pose estimation method for non-cooperative rendezvous. *Appl. Opt.* **2018**, *57*, 6164–6173. [[CrossRef](#)] [[PubMed](#)]
11. Deng, Y.; Xian, N.; Duan, H. A Binocular Vision-based Measuring System for UAVs Autonomous Aerial Refueling. In Proceedings of the IEEE International Conference on Control and Automation (ICCA), Kathmandu, Nepal, 1–3 June 2016.
12. Zhuang, L.; Han, Y.; Fan, Y.; Cao, Y.; Wang, B.; Zhang, Q. Method of pose estimation for UAV landing. *Chin. Opt. Lett.* **2012**, *10*, S20401. [[CrossRef](#)]
13. Benini, A.; Rutherford, M.J.; Valavanis, K.P. Real-time GPU-based Pose Estimation of a UAV for Autonomous Takeoff and Landing. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
14. Tai, J.M.; Fieguth, P.W. Incremental Shape Reconstruction using Stereo Image Sequences. In Proceedings of the International Conference on Image Processing, Vancouver, BC, Canada, 10–13 September 2000.
15. Alix, D.; Walli, K.; Raquet, J. Error characterization of flight trajectories reconstructed using Structure from Motion. In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 14–16 October 2014.
16. Huang, Y.P.; Sithole, L.; Lee, T.T. Structure from motion technique for scene detection using autonomous drone navigation. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, 1–12. [[CrossRef](#)]
17. Schneider, J.; Eling, C.; Klingbeil, L.; Kuhlmann, H.; Förstner, W.; Stachniss, C. Fast and effective online pose estimation and mapping for UAVs. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
18. Andert, F.; Mejias, L. Improving monocular SLAM with altimeter hints for fixed-wing aircraft navigation and emergency landing. In Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.
19. Mary, A.; Gerhard, H. Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter. *Sensors* **2017**, *17*, 2164. [[CrossRef](#)]
20. Konovalenko, I.; Kuznetsova, E.; Miller, A.; Miller, B.; Popov, A.; Shepelev, D.; Stepanyan, K. New Approaches to the Integration of Navigation Systems for Autonomous Unmanned Vehicles (UAV). *Sensors* **2018**, *18*, 3010. [[CrossRef](#)] [[PubMed](#)]
21. Hmam, H.; Kim, J. Aircraft recognition and pose estimation. In Proceedings of the Visual Communications and Image Processing, Perth, Australia, 20–23 June 2000.
22. Wang, L.; Xing, C.; Yan, J. Aircraft pose estimation based on mathematical morphological algorithm and Radon transform. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, Shanghai, China, 26–28 July 2011.
23. Breuers, M.; Reus, N. Image-based aircraft pose estimation: A comparison of simulations and real-world data. In Proceedings of the Automatic target recognition XI, Orlando, FL, USA, 17–20 April 2001.
24. Fu, T.; Sun, X. The relative pose estimation of aircraft based on contour model. In Proceedings of the International Conference on Optical and Photonics Engineering, Xi'an, China, 14–17 October 2016.
25. Wang, X.; Yu, H.; Feng, D. Pose estimation in runway end safety area using geometry structure features. *Aeronaut. J.* **2016**, *120*, 675–691. [[CrossRef](#)]
26. Yuan, W.; Peng, J.; Wang, L.; Lin, S. Aircraft Pose Recognition Using Locally Linear Embedding. In Proceedings of the International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, 11–12 April 2009.
27. Luo, J.; Teng, X.; Zhang, X.; Zhong, L. Structure extraction of straight wing aircraft using consistent line clustering. In Proceedings of the International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017.
28. Sadraey, M.H. *Aircraft Design: A Systems Engineering Approach*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013; pp. 160–249, ISBN 9781119953401.
29. Kholish, R.K.; Aditya, P.; Mochammad, A.M. Design of high altitude long endurance UAV: Structural analysis of composite wing using finite element method. *J. Phys. Conf. Ser.* **2018**, *1005*, 012025. [[CrossRef](#)]
30. Park, K.; Han, J.W.; Lim, H.J.; Kim, B.S.; Lee, J. Optimal design of airfoil with high aspect ratio in unmanned aerial vehicles. *Int. J. Aerosp. Mech. Eng.* **2008**, *2*, 381–387. [[CrossRef](#)]
31. Gioi, R.G.; Jakubowicz, J.; Morel, J.; Randall, G. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 722–732. [[CrossRef](#)] [[PubMed](#)]

32. Zhang, Y.; Liu, Y.; Zou, Z. Comparative study of line extraction method based on repeatability. *J. Comput. Inf. Syst.* **2012**, *8*, 10097–10104.
33. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [[CrossRef](#)]
34. Ester, M.; Kriegel, H.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996.
35. Arthur, D.; Vassilvitskii, S. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007.
36. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004; pp. 310–324, ISBN 9780521540513.
37. AUTODESK. Available online: <https://www.autodesk.com/> (accessed on 6 December 2017).



© 2019 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/>).