sensors

Article

UAV Control on the Basis of 3D Landmark Bearing-Only Observations

Simon Karpenko, Ivan Konovalenko, Alexander Miller, Boris Miller * and Dmitry Nikolaev

Received: 15 August 2015; Accepted: 10 November 2015; Published: 27 November 2015
Academic Editors: Felipe Gonzalez Toro and Antonios Tsourdos

Institute for Information Transmission Problems RAS, Bolshoy Karetny per. 19, Build. 1, GSP-4, Moscow 127051, Russia; simon.karpenko@gmail.com (S.K.); konovalenko@iitp.ru (I.K.); amiller@iitp.ru (A.M.); dimonstr@iitp.ru (D.N.)
* Correspondence: bmiller@iitp.ru; Tel.: +7-495-650-4781; Fax: +7-495-650-0579

Abstract: The article presents an approach to the control of a UAV on the basis of 3D landmark observations. The novelty of the work is the usage of the 3D RANSAC algorithm developed on the basis of the landmarks' position prediction with the aid of a modified Kalman-type filter. Modification of the filter based on the pseudo-measurements approach permits obtaining unbiased UAV position estimation with quadratic error characteristics. Modeling of UAV flight on the basis of the suggested algorithm shows good performance, even under significant external perturbations.

Keywords: UAV; visual odometry; projective geometry; video stream; feature points; modified Kalman filter; control

1. Introduction

Modern UAV's navigation systems use the standard elements of INS (inertial navigation systems) along with GPS, which permits correcting the bias and improving the UAV localization, which are necessary for resolving mapping issues, targeting and reconnaissance tasks [1]. The use of computer vision as a secondary or a primary method for autonomous navigation of UAVs has been discussed frequently in recent years, since the classical combination of GPS and INS systems cannot sustain autonomous flight in many situations [2]. UAV autonomous missions usually need so-called data fusion, which is a difficult task, especially for standard INS and vision equipment. It is clear that cameras provide visual information in a different form, inapplicable to UAV direct control, and therefore, one needs an additional on-board memory and special recognition algorithms.

1.1. Visual-Based Navigation Approaches

Several studies have demonstrated the effectiveness of approaches based on motion field estimation and feature tracking for visual odometry [3]. Vision-based methods have been proposed even in the context of autonomous landing management [2]. In [4], visual odometry based on geometric homography was proposed. However, the homography analysis uses only 2D reference points coordinates, though for the evaluation of the current UAV altitude, the 3D coordinates are necessary. All such approaches presume the presence of some recognition system in order to detect the objects nominated in advance. Examples of such objects can be special buildings, crossroads, tops of mountains, and so on. The principal difficulties are the different scale and aspect angles of observed and stored images, which leads to the necessity of huge template libraries in the memory of the UAV control system. Here, one can avoid this difficulty, because of the usage of another approach based on the observation of so-called feature points [5] that are scale and aspect angle invariant. For this purpose, the technology of feature points [6] is used. In [7], the approach based on the coordinate correspondence of the reference points observed by the on-board camera and the reference points on the map loaded into the UAV's memory before the mission start had been suggested. During the flight, these maps are compared to the frame of the land, directly observed with the help of an on-board video camera. As a result, one can detect the current location and orientation without time-error accumulation. These methods are invariant to some transformations, and they are noise-stable, so that predetermined maps can be different in scale, aspect angle, season, luminosity, weather conditions, *etc*. This technology appeared in [8]. The contribution of this work is the usage of a modified unbiased pseudo-measurements filter for bearing-only observations of some reference points with known terrain coordinates.

1.2. Kalman Filter

In order to obtain metric data from visual observations, one needs first to make observations from different positions *i.e.*, *triangulation* and then use nonlinear filtering. However, all nonlinear filters either have unknown bias [9] or are very difficult for on-board implementation, like the Bayesian-type estimation [10,11]. Approaches for position estimation based on bearing-only observations had been analyzed long ago, especially for submarine applications [12] and nowadays for UAV applications [1].

A comparison of different nonlinear filters for bearing-only observations in the issue of ground-based object localization [13] shows that the EKF (extended Kalman filter), the unscented Kalman filter, the particle filter and the pseudo-measurement filter give almost the same level of accuracy, while the pseudo-measurement filter is usually more stable and simple for on-board implementation. This observation is in accordance with older results [12], where all of these filters were compared in the issue of moving object localization. It has been mentioned that all of these filters have bias, which makes their use in data fusion issues rather problematic [14]. The principle requirement for such filters in data fusion is the non-biased estimate with the known mean square characterization of the error. Among the variety of possible filters, the pseudo-measurement filter can be easily modified to satisfy the data fusion demands. The idea of such nonlinear filtering was developed by V.S. Pugachev and I. Sinitsyn in the form of so-called conditionally-optimal filtering [15], which provides the non-biased estimation within the class of linear filters with the minimum mean squared error. In this paper, we develop such a filter (the so-called pseudo-measurement Kalman filter (PKF)) for the UAV position estimation and give the algorithm for path planning along with the reference trajectory under external perturbations and noisy measurements.

1.3. Optical Absolute Positioning

Some known aerospace maps of a terrain in a flight zone are loaded into the aircraft memory before the start of a flight. During the flight, these maps are compared to the frame of the land, directly observed with the help of an on-board video camera. For this purpose, the technology of feature points [6] is used. As a result, one can detect the current location and orientation without time-error accumulation. These methods are invariant to some transformations and are also noise-stable, so that predetermined maps can vary in height, season, luminosity, weather conditions, *etc*. Furthermore, from the moment of the previous plane surveying, the picture of this landscape can be changed due to human and natural activity. All approaches based on the capturing of the objects assigned in advance presume the presence of some on-board recognition system in order to detect and recognize such objects. Here, we avoid this difficulty by using the observation of feature points [5] that are scale and aspect angle invariant. In addition, the modified pseudo-measurements Kalman filtering (PKF) is used for the estimation of UAV positions and the control algorithm.

1.4. Outline of the Approach and the Article Structure

One of the principal parts of this research is an approach to the estimation of the UAV position on the basis of the bearing-only observations. The original filter that uses the idea of pseudo-measurements had been suggested in reference [16] for the case of the azimuth bearing of the terrain objects nominated in advance. In reference [17], this approach had been extended to the case of two angle measurements, namely azimuth and elevation. However, the usage of this approach as a real navigation tool needs huge on-board memory and a sophisticated recognition algorithm, since the template and in-flight observed images, even of the same object, are rather different due to the changes of illuminance, the altitude of flight and the aspect angles. That is why the method based on the observation of feature points looks more attractive for in-flight implementation. In reference [7], an algorithm joining together the feature points approach and modified PKF had been suggested, though for 2D feature point localization, while the more advanced 3D localization had been suggested in references [18,19], which are the shortened versions of the methodology presented in this article.

In this work, we use a computer simulation of a UAV flight and on-board video camera imaging. The simulation program is written in MATLAB. The type of feature points is ASIFT, realized in OpenCV (Python) [20]. Feature points in this model work as in a real flight, because the images for the camera model and for the template images were transformed by projective mapping and created by observations from different satellites.

The next section presents the original RANSAC algorithm for 3D feature point localization. Sections 3 and 4 give the description of PKF, providing the unbiased estimation of the UAV position with the estimate of quadratic errors. Section 5 describes the locally optimal control algorithm for tracking the reference trajectory on the basis of PKF estimation of the UAV position. In Section 6, we give a new approach to the RANSAC robustness with the use of the UAV motion model. Section 7 presents the modeling results, and Section 8 is the conclusions.

2. Random Sample Consensus for Isometry

At every step, the algorithm deals with two images of a 3D landscape. An example of the landscape used for modeling is shown in Figure 1.

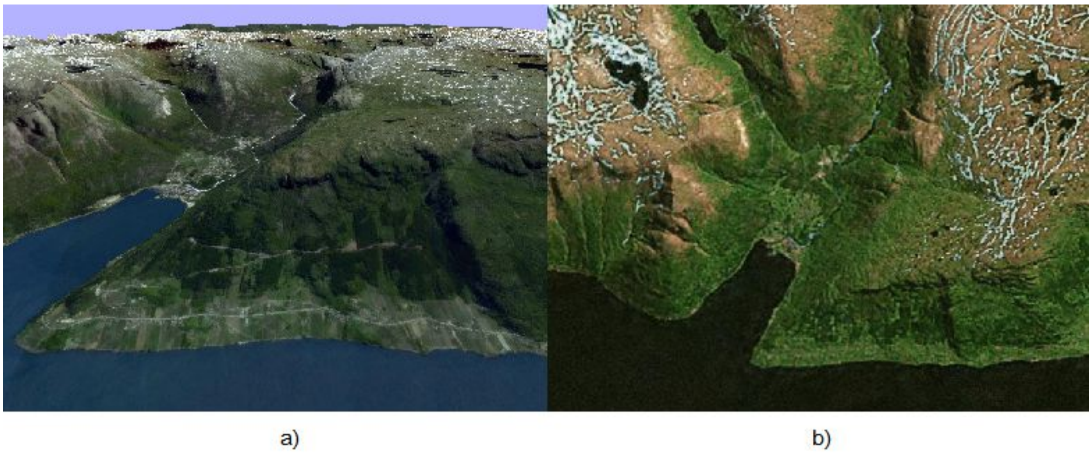

Figure 1. (**a**) Image I_c of the on-board camera of the UAV; (**b**) template image loaded in the UAV memory in advance.

The first image I_c is obtained from the on-board UAV camera, the position of which is unknown and should be estimated. The second image I_m was taken previously from a known position and uploaded into the UAV memory. The ASIFT method is used for both images to detect feature points, which are specified in pixels:

$$c_i = \begin{vmatrix} c_{xi} \\ c_{yi} \end{vmatrix}$$

and:

$$m_i = \begin{vmatrix} m_{xi} \\ m_{yi} \end{vmatrix}$$

and calculates their descriptors. The correspondence between images is constructed by using these descriptors, and thereby, the feature points are combined in pairs. However, many of these pairs are wrong, and therefore, these pairs are considered as outliers or they are not. The result of ASIFT correspondence is shown in Figure 2.

The Earth coordinate system is the Cartesian coordinate system, which is rigidly connected with the Earth. Therefore, the algorithm uses a 3D terrain map of the area from which the image I_m was taken and over which the UAV flies. Therefore, one can determine the coordinates of the points:

$$r_i = \begin{vmatrix} x_i \\ y_i \\ z_i \end{vmatrix}$$

which generated m_i points in the Earth coordinate system. However, if i corresponds to the pair of points that is not an outlier, then the point r_i also generates a point c_i in the UAV camera.

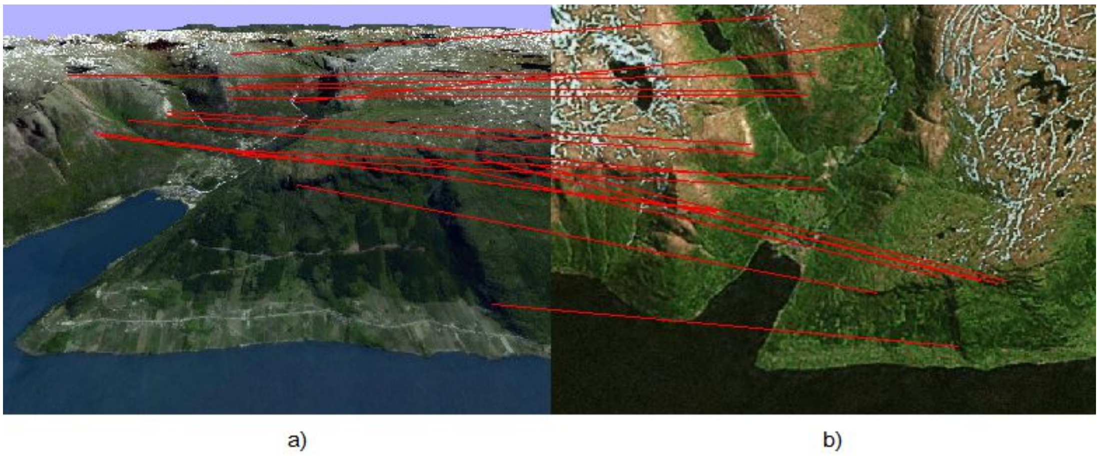

Figure 2. (**a**) Image I_c of the on-board camera of the UAV; (**b**) template image loaded in the UAV memory in advance.

Another Cartesian coordinate system is rigidly connected with the UAV camera. The axis of the camera is parallel to the axis z. The transformation from the Earth coordinate system to the UAV coordinate system has the form:

$$r' = A(r - b)$$

where b represents the coordinates of the camera in the Earth coordinate system and A is the orthogonal ($AA^T = I$) rotation matrix defining the orientation of the UAV camera. Then, the points r_i in the camera coordinate system are:

$$r'_i = A(r_i - b)$$

To define the relation between r_i and the feature points c_i, one can use the model of the camera obscura. This model gives a central projection on the plane:

$$c_i = \begin{vmatrix} \frac{\rho_{xi}}{\rho_{zi}} \\ \frac{\rho_{yi}}{\rho_{zi}} \end{vmatrix}$$

where:

$$\begin{vmatrix} \rho_{xi} \\ \rho_{yi} \\ \rho_{zi} \end{vmatrix} = \rho_i = Kr'_i = KA(r_i - b)$$

where K is a known calibration matrix of the camera.

Thus, the task is to estimate A and b on the basis of known c_i, r_i, K. The minimum number of feature point pairs needed to solve this task is three.

One can give the solution of the problem under the assumption that there are just three pairs:

$$i = \{1,2,3\}$$

Points r'_i form a triangle in the space with the following sides:

$$\rho_1 = ||r'_2 - r'_3||_2, \rho_2 = ||r'_3 - r'_1||_2, \rho_3 = ||r'_1 - r'_2||_2$$

Meanwhile, due to the rectilinear propagation of light, each point r'_i lies on the beam $r' = a_i t$, where t is a scalar parameter, and:

$$a_i = K^{-1} \begin{vmatrix} c_{xi} \\ c_{yi} \\ 1 \end{vmatrix}$$

In order to find r'_i, we have to determine parameters t_i, $i = \{1,2,3\}$ that satisfy the system of quadratic equations:

$$\begin{cases} (a_2t_2 - a_3t_3)^T(a_2t_2 - a_3t_3) = \rho_1^2 \\ (a_3t_3 - a_1t_1)^T(a_3t_3 - a_1t_1) = \rho_2^2 \\ (a_1t_1 - a_2t_2)^T(a_1t_1 - a_2t_2) = \rho_3^2 \end{cases}$$

For the given t_1, this system may be either solved analytically or has no solution. A determination of t_1 can be done numerically, for example by the bisection method.

Finally, one can obtain the coordinates of three points on the Earth's surface in the camera coordinate system r'_i and, at the same time, in the Earth coordinate system r_i. The connection between them is: $r'_i = A(r_i - b)$. Since A is the orthogonal matrix, then $y = Ax$ implies $||y||_2 = ||x||_2$; thereby:

$$r_i'^T r'_i = (r_i - b)^T(r_i - b)$$

Therefore, we have eliminated A and obtained the problem of finding the intersection of three spheres, which can be solved analytically. This problem may have two solutions; one of them will be rejected later. When b has been found, solutions for A are as follows:

$$A = \begin{vmatrix} r'_1 & r'_2 & r'_3 \end{vmatrix} \begin{vmatrix} r_1 - b & r_2 - b & r_3 - b \end{vmatrix}^{-1}$$

Therefore, there are two options, and only one of them is correct:

$$\begin{vmatrix} a_{11} \\ a_{21} \\ a_{31} \end{vmatrix} = \begin{vmatrix} a_{12} \\ a_{22} \\ a_{32} \end{vmatrix} \times \begin{vmatrix} a_{13} \\ a_{23} \\ a_{33} \end{vmatrix} \quad \text{or} \quad \begin{vmatrix} a_{11} \\ a_{21} \\ a_{31} \end{vmatrix} = - \begin{vmatrix} a_{12} \\ a_{22} \\ a_{32} \end{vmatrix} \times \begin{vmatrix} a_{13} \\ a_{23} \\ a_{33} \end{vmatrix}$$

If the first one is correct, then the second one corresponds exactly to the turnover.

As a result, A and b have been found by using three pairs of feature points and the height map. However, this approach alone is not suitable as the final solution, due to the following problems:

1. The method gives either knowingly false solution or no solution at all if among the three points there are outliers.
2. There is a strong dependence on the noise in the feature points' location.

Both problems may be solved with random sample consensus (RANSAC) [21,22]. From the general selection of points, one needs to select N times a subsample of size three. For each subsample $j = \{1, 2, \cdots, N\}$, one can calculate A_j and b_j, which allows one to simulate the generation of all feature points on the UAV camera:

$$c_{ji} = \left| \begin{matrix} \frac{\rho_{jxi}}{\rho_{jzi}} \\ \frac{\rho_{jyi}}{\rho_{jzi}} \end{matrix} \right|, \quad \rho_j = Kr'_{ji} = KA_j(r_{ji} - b_j)$$

Then, one can evaluate which points are the outliers by the threshold $s_{ji} = \mathbf{1}(||c_i - c_{ji}||_2 < d)$, where d is the threshold. Here, $s_{ji} = 0$ means that projection of the i-th point on the j-th point is counted as an outlier, otherwise $s_{ji} = 1$. The answer will be the following:

$$A = A_J, \quad b = b_J, \quad J = \arg\max_j \sum_i s_{ji}$$

Therefore, we really solve the problem of outliers. Next, we find the required number of N subsamples of a size of three, such that among them, there will be at least one subsample without outliers with probability p. Let the proportion of outliers be $1 - w$. It is easy to see that [21]:

$$N(p) = \frac{\log(1 - p)}{\log(1 - w^3)}$$

In the case when $w = \frac{1}{2}$: $N(0.9999) \approx 69$, which shows the high efficiency of algorithm. After that, the points marked as outliers are removed from consideration. The clarification of the response is made by the numerical solution of the following optimization problem on the set of remaining points:

$$\{A^*, b^*\} = \arg\min_{A,b} \sum_i ||c_i - c_i(A, b)||_2^2$$

Thereby, the second problem of noise reduction may be solved. However, one can use a more advanced procedure, which takes into account the motion model. A more stable solution may be obtained with the aid of so-called robust RANSAC [23]; the idea is to use predicted values of (A, b) for the preliminary rejection of outliers from the pairs of observed feature points. Therefore, if on the k-th step of the filtering procedure, the values (A_k, b_k) have been obtained, one can use the following values on the $(k + 1)$-th step:

$$(A_{k+1}, b_{k+1}) = (A_k, \hat{b}_{k+1}) \tag{1}$$

where $\hat{b}_{k+1}$ is the predicted estimate of the UAV attitude obtained on the basis of the PKF estimate. The filtering approach is described in Section 5.

3. Filtering Problem Statement

The problem of bearing-only filtering is considered to determine the coordinates of the UAV, which can observe some objects with known coordinates. These objects can be either the well recognizable objects or a network of radio-beacon stations with a well-specified frequency and known coordinates. In this work, the function of beacons is performed by the feature points determined with the aid of the RANSAC algorithm. The UAV has the standard set of INS devices, which enables it to perform the flight with some degree of accuracy, which, however, is not sufficient for mission completion.

3.1. Model of the UAV's Motion

We assume that a UAV motion is described by three coordinates $(X(t_k), Y(t_k), Z(t_k))$ and velocities $(V_x(t_k), V_y(t_k), V_z(t_k))$. At times $t_k = k\Delta t, \quad k = 1, 2, ...,$ these coordinates satisfy the following equations:

$$\mathbf{X}(t_{k+1}) = F\mathbf{X}(t_k) + B\mathbf{a}(t_k) + \mathbf{W}(t_k) \tag{2}$$

where:

$$\mathbf{X}(t_k) = (X(t_k), Y(t_k), Z(t_k), V_x(t_k), V_y(t_k), V_z(t_k))^T$$

is the vector of state-velocities,

$$\mathbf{a}(t_k) = (a_x(t_k), a_y(t_k), a_z(t_k))^T$$

is the vector of accelerations, which we consider as controls,

$$\mathbf{W}(t_k) = (0, 0, 0, W_x(t_k), W_y(t_k), W_z(t_k))^T$$

is the vector of current perturbations, modeling the turbulence components of the wind and the autopilot errors, as well. The matrices A and B are equal:

$$F = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{pmatrix}$$

and stochastic Equation (2) describes a controlled and perturbed UAV motion.

3.2. Measurements

Assume that (X_i, Y_i, Z_i) are the coordinates of the i-th reference point and $\phi_i(t_k), \lambda_i(t_k)$ are the bearing angles on that point. The measuring scheme is shown in Figure 3.

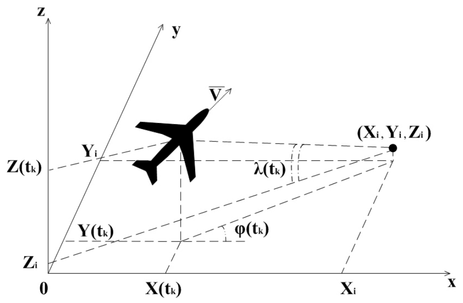

Figure 3. Scheme of the measurements of bearing angles. X_i, Y_i, Z_i are the coordinates of the i-th feature point; $\lambda(t_k), \phi(t_k)$ are the elevation and azimuth bearing angles, measured at the moment t_k.

At moment t_k, these angles satisfy the relations:

$$\begin{aligned}
&\frac{Y_i + \varepsilon_k^y - Y(t_k)}{X_i + \varepsilon_k^x - X(t_k)} I_i(t_k) = \tan\phi_i(t_k) + \varepsilon_k^\phi \\
&\frac{Z_i + \varepsilon_k^z - Z(t_k)}{\sqrt{(X_i + \varepsilon_k^x - X(t_k))^2 + (Y_i + \varepsilon_k^y - Y(t_k))^2}} I_i(t_k) = \tan\lambda_i(t_k) + \varepsilon_k^\lambda
\end{aligned} \tag{3}$$

where $\varepsilon_k^x \sim \mathcal{WN}(0,\sigma_x^2)$, $\varepsilon_k^y \sim \mathcal{WN}(0,\sigma_y^2)$, $\varepsilon_k^z \sim \mathcal{WN}(0,\sigma_z^2)$, $\varepsilon_k^\phi \sim \mathcal{WN}(0,\sigma_\phi^2)$, $\varepsilon_k^\lambda \sim \mathcal{WN}(0,\sigma_\lambda^2)$ are uncorrelated random variables with zero means and variances σ_x^2, σ_y^2, σ_z^2, $\sigma_\phi^2, \sigma_\lambda^2$, defined as errors in the measurement of the coordinates of the i-th reference point and of the tangents of angles $\phi_i(t_k), \lambda_i(t_k)$ and forming the white noise sequences.

Remark 1. *In the majority of works based on the method of pseudo-measurements, another model is used. It assumes the measurements of the angles with Gaussian errors (see [12,14] and most of the successive works). However, in the real definition of the object position in the image or in the matrix of sensors, the system measures the distance between the object image and the center of the sensor, that is the tangent of the bearing angle. This simple observation allows one to find the unbiased estimate of the UAV coordinates.*

One can rewrite Equation (3) for angle $\lambda_i(t_k)$ as follows:

$$\frac{Z_i + \varepsilon_k^z - Z(t_k)}{Y_i + \varepsilon_k^y - Y(t_k)} \sin\phi_i(t_k) I_i(t_k) = \frac{\sin\lambda_i(t_k)}{\cos\lambda_i(t_k)} + \varepsilon_k^\lambda \tag{4}$$

Remark 2. *The indicator function $I_i(t_k) = 1$ if at t_k the bearing of the i-th reference point occurs, and $I_i(t_k) = 0$ otherwise. For convenience, we assume that $I_i(t_k) = 1$.*

Therefore, at the moment t_k, the UAV control system determines the angles $\phi_i(t_k)$ and $\lambda_i(t_k)$, related to the coordinates of the UAV $(X(t_k), Y(t_k), Z(t_k))$ as follows:

$$\begin{aligned}
&(Y_i + \varepsilon_k^y - Y(t_k))\cos\phi_i(t_k) - (X_i + \varepsilon_k^x - X(t_k))\sin\phi_i(t_k) \\
&= \varepsilon_k^\phi (X_i + \varepsilon_k^x - X(t_k))\cos\phi_i(t_k) \\
&(Z_i + \varepsilon_k^z - Z(t_k))\sin\phi_i(t_k)\cos\lambda_i(t_k) - (Y_i + \varepsilon_k^y - Y(t_k))\sin\lambda_i(t_k) \\
&= \varepsilon_k^\lambda (Y_i + \varepsilon_k^y - Y(t_k))\cos\lambda_i(t_k)
\end{aligned} \tag{5}$$

4. Modified Kalman Filtering on the Basis of Pseudo-Measurements

4.1. Linear Measurements Model

The idea of the pseudo-measurement method is to separate in Equation (5) the observable and non-observable values, which gives the following observation equations:

$$\begin{aligned}
&m_k^\phi = Y_i\cos\phi_i(t_k) - X_i\sin\phi_i(t_k) = Y(t_k)\cos\phi_i(t_k) - X(t_k)\sin\phi_i(t_k) \\
&-\varepsilon_k^y\cos\phi_i(t_k) + \varepsilon_k^x\sin\phi_i(t_k) + \varepsilon_k^\phi(X_i + \varepsilon_k^x - X(t_k))\cos\phi_i(t_k) \\
&m_k^\lambda = Z_i\sin\phi_i(t_k)\cos\lambda_i(t_k) - Y_i\sin\lambda_i(t_k) = Z(t_k)\sin\phi_i(t_k)\cos\lambda_i(t_k) - Y(t_k)\sin\lambda_i(t_k) \\
&-\varepsilon_k^z\sin\phi_i(t_k)\cos\lambda_i(t_k) + \varepsilon_k^y\sin\lambda_i(t_k) + \varepsilon_k^\lambda(Y_i + \varepsilon_k^y - Y(t_k))\cos\lambda_i(t_k)
\end{aligned} \tag{6}$$

where X_i, Y_i, Z_i represent the coordinates of the i-th feature point determined with the aid of the RANSAC algorithm and ϕ_i, λ_i are the corresponding observable bearing angles measured by the system. Thus, the left-hand side of Equation (6), that is (m_k^ϕ, m_k^λ), corresponds to the observable values, whereas the right-hand side containing the coordinates of the UAV corresponds to the non-observable ones. The aim is to estimate the coordinates and velocities of the UAV on the basis of linear observation Equation (6) and the motion model Equation (2). Therefore, the measurement vector has the following form:

$$m_k = \begin{pmatrix} m_k^{\phi} \\ m_k^{\lambda} \end{pmatrix} = \begin{pmatrix} Y(t_k)\cos\phi_i(t_k) - X(t_k)\sin\phi_i(t_k) - \varepsilon_k^y\cos\phi_i(t_k) + \varepsilon_k^x\sin\phi_i(t_k) \\ +\varepsilon_k^{\phi}(X_i + \varepsilon_k^x - X(t_k))\cos\phi_i(t_k) \\ \\ Z(t_k)\sin\phi_i(t_k)\cos\lambda_i(t_k) - Y(t_k)\sin\lambda_i(t_k) - \varepsilon_k^z\sin\phi_i(t_k)\cos\lambda_i(t_k) \\ +\varepsilon_k^y\sin\lambda_i(t_k) + \varepsilon_k^{\lambda}(Y_i + \varepsilon_k^y - Y(t_k))\cos\lambda_i(t_k) \end{pmatrix} \tag{7}$$

Thereby, we obtain the system Equation (7) of linear measurement equations, though the noise variance depends on non-observable coordinates. By using V.S. Pugachev's method [15], one can obtain the unbiased estimate and the variance with the aid of a prediction-correction filter [24]. Moreover, we do not need to assume the Gaussian distribution of errors that is not valid in bearing observations with optical-electronic cameras with discrete image sensors.

4.2. Prediction-Correction Estimation

Assume that at the moment t_k, we have unbiased estimates $\hat{\mathbf{X}}(t_k)$, such that:

$$E(\hat{\mathbf{X}}(t_k)) = \mathbf{X}(t_k) \tag{8}$$

with the following matrix of the mean-square errors:

$$\hat{P}(t_k) = E\left\{(\hat{\mathbf{X}}(t_k) - \mathbf{X}(t_k))(\hat{\mathbf{X}}(t_k) - \mathbf{X}(t_k))^T\right\}$$
$$= \begin{pmatrix} \hat{p}^{xx}(t_k) & \hat{p}^{xy}(t_k) & \dots & \dots & \dots & \hat{p}^{xV_z}(t_k) \\ \hat{p}^{xy}(t_k) & \hat{p}^{yy}(t_k) & \dots & \dots & \dots & \hat{p}^{yV_z}(t_k) \\ \hat{p}^{xz}(t_k) & \hat{p}^{yz}(t_k) & \dots & \dots & \dots & \hat{p}^{zV_z}(t_k) \\ \hat{p}^{xV_x}(t_k) & \hat{p}^{yV_x}(t_k) & \dots & \dots & \dots & \hat{p}^{V_xV_z}(t_k) \\ \hat{p}^{xV_y}(t_k) & \hat{p}^{yV_y}(t_k) & \dots & \dots & \dots & \hat{p}^{V_yV_z}(t_k) \\ \hat{p}^{xV_z}(t_k) & \hat{p}^{yV_z}(t_k) & \dots & \dots & \dots & \hat{p}^{V_zV_z}(t_k) \end{pmatrix} \tag{9}$$

Problem 1. *Find the unbiased estimates $\hat{\mathbf{X}}(t_{k+1})$ and matrix $\hat{P}(t_{k+1})$ on the basis of estimates at the moment t_k, m_k, the known position of the i-th observable object (X_i, Y_i, Z_i) and the UAV's motion parameter Equation (2). These estimates must satisfy Equation (8) and give the matrix Equation (9) for the moment t_{k+1}.*

4.2.1. Prediction

The prediction is obtained by assuming that at the moment t_{k+1}, the values of $\phi(t_{k+1})$, $\lambda(t_{k+1})$ will be known:

$$\tilde{\mathbf{X}}(t_{k+1}) = F\hat{\mathbf{X}}(t_k) + B\mathbf{a}(t_k)$$
$$\tilde{m}_{k+1} = \begin{pmatrix} \tilde{m}_{k+1}^{\phi} \\ \tilde{m}_{k+1}^{\lambda} \end{pmatrix} = \begin{pmatrix} I(t_{k+1})(\tilde{Y}(t_{k+1})\cos\phi(t_{k+1}) - \tilde{X}(t_{k+1})\sin\phi(t_{k+1})) \\ I(t_{k+1})(\tilde{Z}(t_{k+1})\sin\phi(t_{k+1})\cos\lambda(t_{k+1}) - \tilde{Y}(t_{k+1})\sin\lambda(t_{k+1})) \end{pmatrix} \tag{10}$$

Assuming that the motion perturbations and the UAV position are uncorrelated, we obtain:

$$\tilde{P}(t_{k+1}) = \begin{pmatrix} \tilde{P}^{xx}(t_{k+1}) & \tilde{P}^{xy}(t_{k+1}) & ... & ... & ... & \tilde{P}^{xV_z}(t_{k+1}) \\ \tilde{P}^{xy}(t_{k+1}) & \tilde{P}^{yy}(t_{k+1}) & ... & ... & ... & \tilde{P}^{yV_z}(t_{k+1}) \\ \tilde{P}^{xz}(t_{k+1}) & \tilde{P}^{yz}(t_{k+1}) & ... & ... & ... & \tilde{P}^{zV_z}(t_{k+1}) \\ \tilde{P}^{xV_x}(t_{k+1}) & \tilde{P}^{yV_x}(t_{k+1}) & ... & ... & ... & \tilde{P}^{V_xV_z}(t_{k+1}) \\ \tilde{P}^{xV_y}(t_{k+1}) & \tilde{P}^{yV_y}(t_{k+1}) & ... & ... & ... & \tilde{P}^{V_yV_z}(t_{k+1}) \\ \tilde{P}^{xV_z}(t_{k+1}) & \tilde{P}^{yV_z}(t_{k+1}) & ... & ... & ... & \tilde{P}^{V_zV_z}(t_{k+1}) \end{pmatrix} \tag{11}$$

where the elements of this matrix are:

$$\begin{aligned}
\tilde{P}^{xx}(t_{k+1}) &= \hat{P}^{xx}(t_k) + 2\hat{P}^{xV_x}(t_k)\Delta t + \hat{P}^{V_xV_x}(t_k)\Delta t^2 \\
\tilde{P}^{yy}(t_{k+1}) &= \hat{P}^{yy}(t_k) + 2\hat{P}^{yV_y}(t_k)\Delta t + \hat{P}^{V_yV_y}(t_k)\Delta t^2 \\
\tilde{P}^{zz}(t_{k+1}) &= \hat{P}^{zz}(t_k) + 2\hat{P}^{zV_z}(t_k)\Delta t + \hat{P}^{V_zV_z}(t_k)\Delta t^2 \\
\tilde{P}^{xy}(t_{k+1}) &= \hat{P}^{xy}(t_k) + \hat{P}^{xV_y}(t_k)\Delta t + \hat{P}^{yV_x}(t_k)\Delta t + \hat{P}^{V_xV_y}(t_k)\Delta t^2 \\
\tilde{P}^{xz}(t_{k+1}) &= \hat{P}^{xz}(t_k) + \hat{P}^{xV_z}(t_k)\Delta t + \hat{P}^{zV_x}(t_k)\Delta t + \hat{P}^{V_xV_z}(t_k)\Delta t^2 \\
\tilde{P}^{yz}(t_{k+1}) &= \hat{P}^{yz}(t_k) + \hat{P}^{yV_z}(t_k)\Delta t + \hat{P}^{zV_y}(t_k)\Delta t + \hat{P}^{V_yV_z}(t_k)\Delta t^2 \\
\tilde{P}^{xV_x}(t_{k+1}) &= \hat{P}^{xV_x}(t_k) + \hat{P}^{V_xV_x}(t_k)\Delta t \\
\tilde{P}^{xV_y}(t_{k+1}) &= \hat{P}^{xV_y}(t_k) + \hat{P}^{V_xV_y}(t_k)\Delta t \\
\tilde{P}^{xV_z}(t_{k+1}) &= \hat{P}^{xV_z}(t_k) + \hat{P}^{V_xV_z}(t_k)\Delta t \\
\tilde{P}^{yV_x}(t_{k+1}) &= \hat{P}^{yV_x}(t_k) + \hat{P}^{V_xV_y}(t_k)\Delta t \\
\tilde{P}^{yV_y}(t_{k+1}) &= \hat{P}^{yV_y}(t_k) + \hat{P}^{V_yV_y}(t_k)\Delta t \\
\tilde{P}^{yV_z}(t_{k+1}) &= \hat{P}^{yV_z}(t_k) + \hat{P}^{V_yV_z}(t_k)\Delta t \\
\tilde{P}^{zV_x}(t_{k+1}) &= \hat{P}^{zV_x}(t_k) + \hat{P}^{V_xV_z}(t_k)\Delta t \\
\tilde{P}^{zV_y}(t_{k+1}) &= \hat{P}^{zV_y}(t_k) + \hat{P}^{V_yV_z}(t_k)\Delta t \\
\tilde{P}^{zV_z}(t_{k+1}) &= \hat{P}^{zV_z}(t_k) + \hat{P}^{V_zV_z}(t_k)\Delta t \\
\tilde{P}^{V_xV_x}(t_{k+1}) &= \hat{P}^{V_xV_x}(t_k) + \sigma^2_{\mathbf{X}} \\
\tilde{P}^{V_yV_y}(t_{k+1}) &= \hat{P}^{V_yV_y}(t_k) + \sigma^2_{\mathbf{Y}} \\
\tilde{P}^{V_zV_z}(t_{k+1}) &= \hat{P}^{V_zV_z}(t_k) + \sigma^2_{\mathbf{Z}} \\
\tilde{P}^{V_xV_y}(t_{k+1}) &= \hat{P}^{V_xV_y}(t_k) \\
\tilde{P}^{V_xV_z}(t_{k+1}) &= \hat{P}^{V_xV_z}(t_k) \\
\tilde{P}^{V_yV_z}(t_{k+1}) &= \hat{P}^{V_yV_z}(t_k)
\end{aligned} \tag{12}$$

Note that $\sigma^2_{\mathbf{X}}$ is not the same as σ^2_x and similarly for the other indices.

Then, the following values $\tilde{P}^{xm}(t_{k+1})$, $\tilde{P}^{ym}(t_{k+1})$, $\tilde{P}^{zm}(t_{k+1})$, $\tilde{P}^{V_xm}(t_{k+1})$, $\tilde{P}^{V_ym}(t_{k+1})$, $\tilde{P}^{V_zm}(t_{k+1})$, $\tilde{P}^{mm}(t_{k+1})$ should be calculated using the following relations:

$$\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} (Y(t_{k+1}) - \tilde{Y}(t_{k+1}))\cos\phi_i(t_{k+1}) - (X(t_{k+1}) - \tilde{X}(t_{k+1}))\sin\phi_i(t_{k+1}) \\ -\varepsilon^y_{k+1}\cos\phi_i(t_{k+1}) + \varepsilon^x_{k+1}\sin\phi_i(t_{k+1}) + \varepsilon^{\phi}_{k+1}(X_i + \varepsilon^x_{k+1} - X(t_{k+1}))\cos\phi_i(t_{k+1}) \\ (Z(t_{k+1}) - \tilde{Z}(t_{k+1}))\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - (Y(t_{k+1}) - \tilde{Y}(t_{k+1}))\sin\lambda_i(t_{k+1}) \\ -\varepsilon^z_{k+1}\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) + \varepsilon^y_{k+1}\sin\lambda_i(t_{k+1}) + \varepsilon^{\lambda}_{k+1}(Y_i + \varepsilon^y_{k+1} - Y(t_{k+1}))\cos\lambda_i(t_{k+1}) \end{pmatrix}$$

and the identities:

$$\begin{aligned}
X_i - X(t_{k+1}) &= X_i - \tilde{X}(t_{k+1}) - (X(t_{k+1}) - \tilde{X}(t_{k+1})) \\
Y_i - Y(t_{k+1}) &= Y_i - \tilde{Y}(t_{k+1}) - (Y(t_{k+1}) - \tilde{Y}(t_{k+1})) \\
Z_i - Z(t_{k+1}) &= Z_i - \tilde{Z}(t_{k+1}) - (Z(t_{k+1}) - \tilde{Z}(t_{k+1}))
\end{aligned}$$

where we consider that the position of the *i*-th object is known and use the non-correlatedness of ε^x_{k+1}, ε^y_{k+1}, ε^z_{k+1}, ε^{ϕ}_{k+1}, $\varepsilon^{\lambda}_{k+1}$ and differences $(X(t_{k+1}) - \tilde{X}(t_{k+1}))$, $(Y(t_{k+1}) - \tilde{Y}(t_{k+1}))$ and $(Z(t_{k+1}) - \tilde{Z}(t_{k+1}))$.

Finally, we get:

$$\begin{aligned}\left[\tilde{P}^{xm}(t_{k+1})\right]^T &= E\left[(X(t_{k+1}) - \tilde{X}(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{xy}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xx}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{xz}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{xy}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix},\end{aligned} \tag{13}$$

$$\begin{aligned}\left[\tilde{P}^{ym}(t_{k+1})\right]^T &= E\left[(Y(t_{k+1}) - \tilde{Y}(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{yy}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xy}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{yz}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{yy}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix},\end{aligned} \tag{14}$$

$$\begin{aligned}\left[\tilde{P}^{zm}(t_{k+1})\right]^T &= E\left[(Z(t_{k+1}) - \tilde{Z}(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{yz}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xz}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{zz}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{yz}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix},\end{aligned} \tag{15}$$

$$\begin{aligned}\left[\tilde{P}^{V_x m}(t_{k+1})\right]^T &= E\left[(V_x(t_{k+1}) - \tilde{V}_x(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{yV_x}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xV_x}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{zV_x}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{yV_x}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix},\end{aligned} \tag{16}$$

$$\begin{aligned}\left[\tilde{P}^{V_y m}(t_{k+1})\right]^T &= E\left[(V_y(t_{k+1}) - \tilde{V}_y(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{yV_y}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xV_y}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{zV_y}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{yV_y}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix},\end{aligned} \tag{17}$$

$$\begin{aligned}\left[\tilde{P}^{V_z m}(t_{k+1})\right]^T &= E\left[(V_z(t_{k+1}) - \tilde{V}_z(t_{k+1}))\begin{pmatrix} m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1} \\ m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1} \end{pmatrix}\right] \\ &= \begin{pmatrix} \tilde{P}^{yV_z}(t_{k+1})\cos\phi_i(t_{k+1}) - \tilde{P}^{xV_z}(t_{k+1})\sin\phi_i(t_{k+1}) \\ \tilde{P}^{zV_z}(t_{k+1})\sin\phi_i(t_{k+1})\cos\lambda_i(t_{k+1}) - \tilde{P}^{yV_z}(t_{k+1})\sin\lambda_i(t_{k+1}) \end{pmatrix}.\end{aligned} \tag{18}$$

In a similar way, we calculate:

$$\begin{aligned}&(\tilde{P}^{mm}(t_{k+1}))^{-1} \\ &= \left[E\begin{pmatrix} (m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1})^2 & (m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1})(m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1}) \\ (m^{\phi}_{k+1} - \tilde{m}^{\phi}_{k+1})(m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1}) & (m^{\lambda}_{k+1} - \tilde{m}^{\lambda}_{k+1})^2 \end{pmatrix}\right]^{-1}\end{aligned} \tag{19}$$

Therefore:

$$\begin{aligned}
&E\left[(m^{\phi}_{k+1}-\tilde{m}^{\phi}_{k+1})^2\right]\\
&=\tilde{P}^{yy}(t_{k+1})\cos^2\phi_i(t_{k+1})-\tilde{P}^{xy}(t_{k+1})\sin 2\phi_i(t_{k+1})\\
&+\tilde{P}^{xx}(t_{k+1})\sin^2\phi_i(t_{k+1})+\sigma^2_y\cos^2\phi_i(t_{k+1})+\sigma^2_x\sin^2\phi_i(t_{k+1})+\sigma^2_{\phi}((X_i-\hat{X}(t_k))^2\\
&+\tilde{P}^{xx}(t_{k+1})+\sigma^2_x)\cos^2\phi_i(t_{k+1})
\end{aligned}$$

$$\begin{aligned}
&E\left[(m^{\phi}_{k+1}-\tilde{m}^{\phi}_{k+1})(m^{\lambda}_{k+1}-\tilde{m}^{\lambda}_{k+1})\right]\\
&=\tilde{P}^{yz}(t_{k+1})\sin\phi_i(t_{k+1})\cos\phi_i(t_{k+1})\cos\lambda_i(t_{k+1})\\
&-\tilde{P}^{yy}(t_{k+1})\cos\phi_i(t_{k+1})\sin\lambda_i(t_{k+1})-\tilde{P}^{xz}(t_{k+1})\sin^2\phi_i(t_{k+1})\cos\lambda_i(t_{k+1})\\
&+\tilde{P}^{xy}(t_{k+1})\sin\phi_i(t_{k+1})\sin\lambda_i(t_{k+1})
\end{aligned}$$

$$\begin{aligned}
&E\left[(m^{\lambda}_{k+1}-\tilde{m}^{\lambda}_{k+1})^2\right]\\
&=\tilde{P}^{zz}(t_{k+1})\sin^2\phi_i(t_{k+1})\cos^2\lambda_i(t_{k+1})-\tilde{P}^{yz}(t_{k+1})\sin\phi_i(t_{k+1})\sin 2\lambda_i(t_{k+1})\\
&+\tilde{P}^{yy}(t_{k+1})\sin^2\lambda_i(t_{k+1})+\sigma^2_z\sin^2\phi_i(t_{k+1})\cos^2\lambda_i(t_{k+1})+\sigma^2_y\sin^2\lambda_i(t_{k+1})+\sigma^2_{\lambda}((Y_i-\hat{Y}(t_k))^2\\
&+\tilde{P}^{yy}(t_{k+1})+\sigma^2_y)\cos^2\lambda_i(t_{k+1})
\end{aligned}$$

4.2.2. Correction

After getting the measurements at the moment t_{k+1}, one can obtain the estimate of the UAV position at this moment. Therefore, the solution of Problem 1 has the form:

$$\hat{\mathbf{X}}(t_{k+1})=\tilde{\mathbf{X}}(t_{k+1})+\tilde{\mathbf{P}}(t_{k+1})(\tilde{P}^{mm}(t_{k+1}))^{-1}(m_{k+1}-\tilde{m}_{k+1}) \tag{20}$$

and the matrix of the mean square errors is equal to:

$$\hat{\mathbf{P}}(t_{k+1})=\tilde{P}(t_{k+1})-\tilde{\mathbf{P}}(t_{k+1})(\tilde{P}^{mm}(t_{k+1}))^{-1}\tilde{\mathbf{P}}(t_{k+1})^T \tag{21}$$

where:

$$\tilde{\mathbf{P}}(t_{k+1})=(\tilde{P}^{xm}(t_{k+1}),\tilde{P}^{ym}(t_{k+1}),...,\tilde{P}^{V_zm}(t_{k+1}))^T$$

5. Robust Filtering on the Basis of the UAV Motion Model

The RANSAC method calculates the rotation matrix and the coordinates of the camera $\{A^*, b^*\}$ in the Earth coordinate system with some minor error. However, the RANSAC method can give quite the wrong answer, called an outlier. It could happen, for example, if the frames I_c and I_m do not depict common objects. We provide further a method that makes a decision about whether $\{A^*, b^*\}$ is an outlier or not. This problem has been considered in relation to the exclusion of outliers in the RANSAC-type procedures [25,26]. Here, we use the modification of the robust RANSAC [23] based on PKF for bearing-only observations [17].

After the prediction step of the Kalman filter, one can estimate the UAV (camera) position and the matrix of the mean square errors:

$$\tilde{\mathbf{X}}=\tilde{\mathbf{X}}(\mathbf{t_{k+1}}),\quad \tilde{\mathbf{P}}=\tilde{\mathbf{P}}(\mathbf{t_{k+1}}).$$

Like in [26], one can suppose that the corresponding probability density is Gaussian. The reason is that the PKF gives the best linear estimates obtained until the current time t. This estimate is the sum of uncorrelated random variables with almost the same variations, at least on the short intervals preceding the current time. It gives the opportunity to approximate the probability density distribution by the Gaussian one. Further extension of the robust RANSAC technique is based on the prior distribution of the UAV attitude. The approach has been developed in [23,27] on the basis of the extended Kalman filter (EKF). However, the estimate given by the EKF has an unknown bias and, of course, does not give the posterior Gaussian distribution. Therefore, the PKF, which gives an unbiased estimate, looks more preferable under the hypothesis of the posterior Gaussian distribution. Therefore, at the $(k+1)$-th step, the posterior distribution of r_i' corresponding to an inlier is assumed to be Gaussian, that is according to Equation (1):

$$\mathcal{N}(A_k(r_i - \tilde{\mathbf{X}}(t_{k+1})), A_k(\tilde{\mathbf{P}}(t_{k+1}) + P_{rr})A_k^T)$$

where P_{rr} is the covariance matrix of the landmarks localization. Further, in the estimation algorithm, the pair $\{A_k, \tilde{\mathbf{X}}(t_{k+1})\}$ is considered as an outlier at the confidence level 95% if:

$$\|r_i' - A_k(r_i - \tilde{\mathbf{X}}(t_{k+1}))\| \geq 2 * Sp[A_k(\tilde{\mathbf{P}}(t_{k+1}) + P_{rr})A_k^T]$$

Otherwise, the correction step is based on $\{A_k, \tilde{\mathbf{X}}(t_{k+1})\}$. Of course, all such nonlinear conjectures need confirmation on the basis of statistical modeling, which is one of the results of this article. One can observe the performance of robust filtering in the following figures. Figure 4 shows the correspondence between feature points obtained without a UAV motion model. Next, Figure 5 shows the correspondence established on the basis of the UAV motion model. The number of outliers reduces substantially.

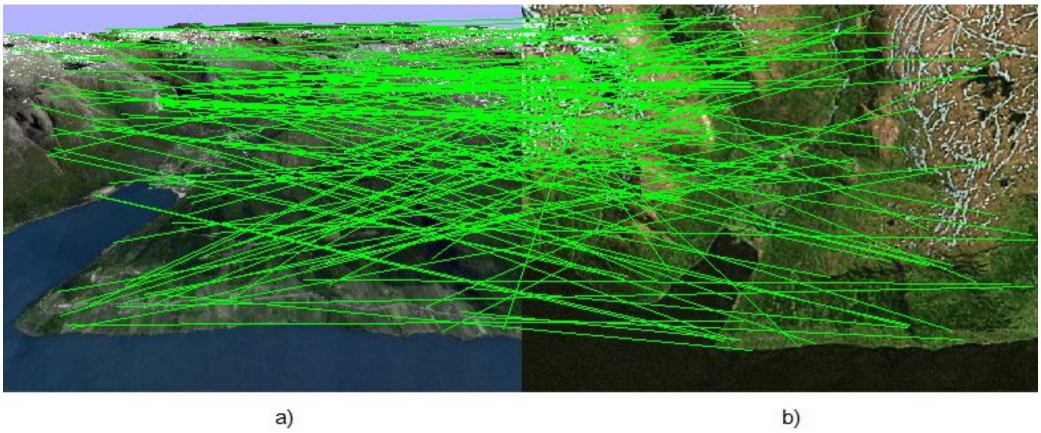

Figure 4. Correspondence between (**a**) (image I_c of the on-board camera of the UAV) and (**b**) (template image loaded in the UAV memory in advance) feature points found without the UAV motion model. One can observe chaotic correspondence, which gives a huge number of outliers.

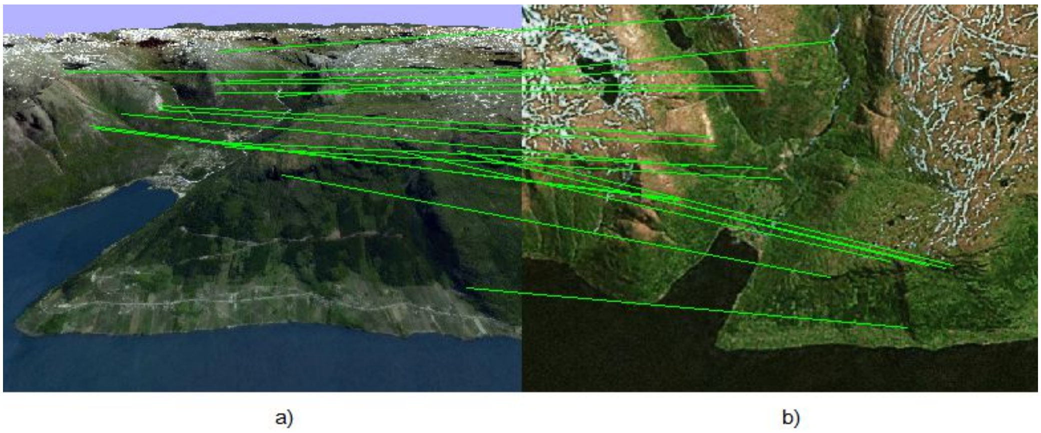

Figure 5. Correspondence between (**a**) (image I_c of the on-board camera of the UAV) and (**b**) (template image loaded in the UAV memory in advance) feature points found with the aid of RANSAC based on the UAV motion model. The number of outliers reduces substantially.

6. Control of the UAV

Control of a UAV that ensures its motion along the reference trajectory may be determined on the basis of the standard deterministic linear-quadratic approach [28]. However, the problem of control on the basis of bearing-only observation is different from the standard one. It should be underlined that this problem is a non-linear one and cannot be solved by the standard way. The problem of the optimal control for the system Equation (2) is the stochastic one with incomplete information and does not have an explicit solution. However, for practical reasons, one can simplify it by considering the locally optimal control. Here, we discuss the following problem:

Problem 2. *Find the locally optimal controls* $a_x(t_k)$, $a_y(t_k)$ *and* $a_z(t_k)$ *aimed to keep the motion of the UAV along the reference trajectory.*

Assume that we have some reference trajectory $\mathbf{X}_{nom}(t_k)$.

Therefore, at the moment t_{k+1}, we have to minimize the following expressions:

$$\begin{aligned}
E_1 &= E\{(X(t_{k+1}) - X_{nom}(t_{k+1}))^2 + (V_x(t_{k+1}) - V_{x_{nom}}(t_{k+1}))\Delta t)^2\} \to \min_{a_x(t_k)} \\
E_2 &= E\{(Y(t_{k+1}) - Y_{nom}(t_{k+1}))^2 + (V_z(t_{k+1}) - V_{z_{nom}}(t_{k+1}))\Delta t)^2\} \to \min_{a_y(t_k)} \\
E_3 &= E\{(Z(t_{k+1}) - Z_{nom}(t_{k+1}))^2 + (V_z(t_{k+1}) - V_{z_{nom}}(t_{k+1}))\Delta t)^2\} \to \min_{a_z(t_k)}
\end{aligned}$$

Let us consider the components of the E_1 expression:

$$\begin{aligned}
&X(t_{k+1}) - X_{nom}(t_{k+1}) = X(t_k) - \hat{X}(t_k) - (X_{nom}(t_k) - \hat{X}(t_k)) \\
&+ (V_x(t_k) - \hat{V}_x(t_k))\Delta t - (V_{x_{nom}}(t_k) - \hat{V}_x(t_k))\Delta t + (a_x(t_k) - a_{x_{nom}}(t_k))\frac{\Delta t^2}{2}
\end{aligned}$$

$$\begin{aligned}
&V_x(t_{k+1}) - V_{x_{nom}}(t_{k+1}) = V_x(t_k) - \hat{V}_x(t_k) - (V_{x_{nom}}(t_k) - \hat{V}_x(t_k)) + (a_x(t_k) - a_{x_{nom}}(t_k))\Delta t \\
&+ W_x(t_k)
\end{aligned}$$

Then, we square these components and take the derivative of the sum with respect to $a_x(t_k)$ given that some components are uncorrelated:

$$\begin{aligned}
E\{(X(t_k) - \hat{X}(t_k))(X_{nom}(t_k) - \hat{X}(t_k))\} = 0 \\
E\{(V_x(t_k) - \hat{V}_x(t_k))(V_{x_{nom}}(t_k) - \hat{V}_x(t_k))\} = 0
\end{aligned}$$

Finally, we get:

$$a_x(t_k) = a_{x_{nom}}(t_k) + \tfrac{2(X_{nom}(t_k) - \hat{X}(t_k))}{5\Delta t^2} + \tfrac{6(V_{x_{nom}}(t_k) - \hat{V}_x(t_k))}{5\Delta t} \quad (22)$$

We take into account that the acceleration of the UAV has limitations $[a_{x_{min}}, a_{x_{max}}]$, so the control acceleration has the form:

$$a_x^c(t_k) = \begin{cases} a_{x_{min}} & if \quad a_x(t_k) < a_{x_{min}}, \\ a_x(t_k) & if \quad a_{x_{min}} \le a_x(t_k) \le a_{x_{max}}, \\ a_{x_{max}} & if \quad a_x(t_k) > a_{x_{max}}. \end{cases}$$

Similarly, we obtain the expressions for $a_y^c(t_k)$ and $a_z^c(t_k)$. Thus, we get the following solution of Problem 2 [19]:

$$\hat{\mathbf{a}}(t_k) = \hat{\mathbf{a}}_{nom}(t_k) + \tfrac{2}{5\Delta t^2}(\hat{\mathbf{X}}_{nom}(t_k) - \hat{\mathbf{X}}(t_k)) + \tfrac{6}{5\Delta t}(\hat{\mathbf{V}}_{nom}(t_k) - \hat{\mathbf{V}}(t_k)) \quad (23)$$

where:

$$\hat{\mathbf{a}}(t_k) = (a_x(t_k), a_y(t_k), a_z(t_k))^T$$
$$\hat{\mathbf{X}}(t_k) = (X(t_k), Y(t_k), Z(t_k))^T$$
$$\hat{\mathbf{V}}(t_k) = (V_x(t_k), V_y(t_k), V_z(t_k))^T$$

and similarly for the nominal trajectory components.

7. Experimental Results

In this section, we give the results of the algorithm's modeling. The UAV is virtually flying over the landscape shown in Figure 1. This image has been obtained from Google Maps, and for verification of the algorithm, the image of the same region obtained from another Bing satellite was used. Therefore, these two images modeled the preliminary downloaded template and the image obtained by the on-board camera. The result of virtual flight experiment is shown in Figure 6.

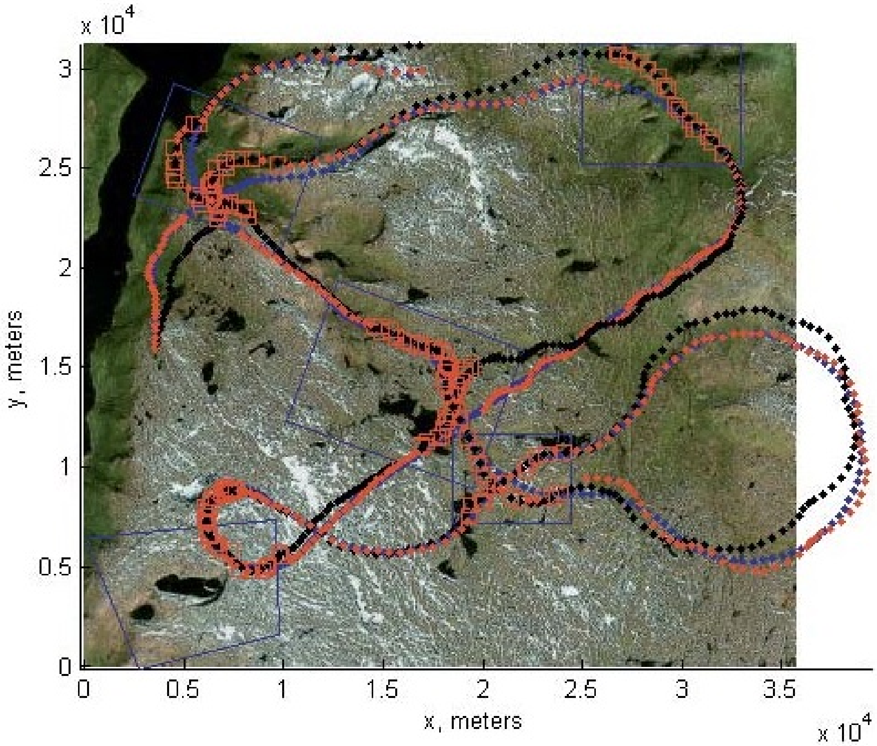

Figure 6. Blue dots corresponds to the reference trajectory and black dots to the real path. Blue squares show the localization of the terrain areas corresponding to the template images, and red squares show the moments where the estimates of the UAV positions have been obtained and assumed to be reliable according to the robust RANSAC algorithm described in Section 5.

8. Results and Discussion

In the modeling of the control algorithm, we use the UAV moving approximately with a velocity of 50 m/s, though the change of the altitude is assumed to be rather substantial. The control algorithm takes into account the constraints imposed on linear acceleration and angular velocities. The software developed for modeling may be used in a real on-board navigation system, as well. Moreover, the filtering algorithm based on unbiased estimation may be easily incorporated with the INS, since it gives also the unbiased square error estimates, which opens the way to correct data fusion. The quality of tracking for x, y, z components is shown below in Figures 7–9, respectively. In all of these figures, blue dots correspond to the reference trajectory, black dots to the real path and red squares to the moments, where the estimates of the UAV positions have been obtained and assumed to be reliable according to the robust RANSAC algorithm. One can observe that in the "measurement" areas, the algorithm estimates the coordinates with high accuracy, and the control provides the tracking with high accuracy, as well.

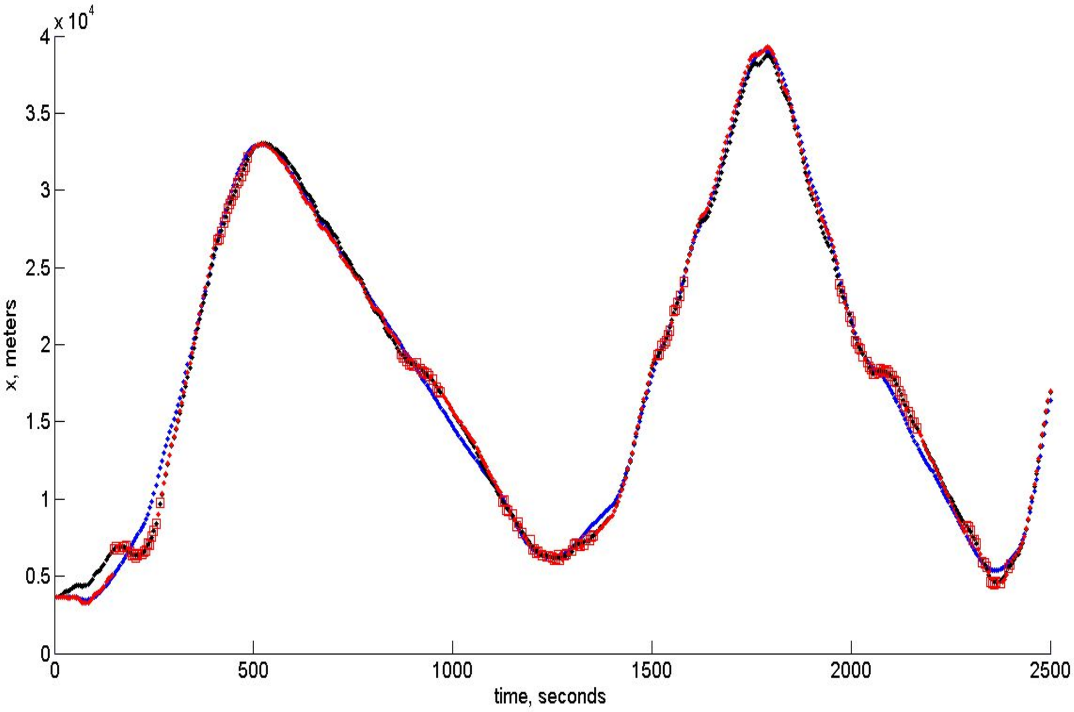

Figure 7. Tracking of the *x*-coordinate.

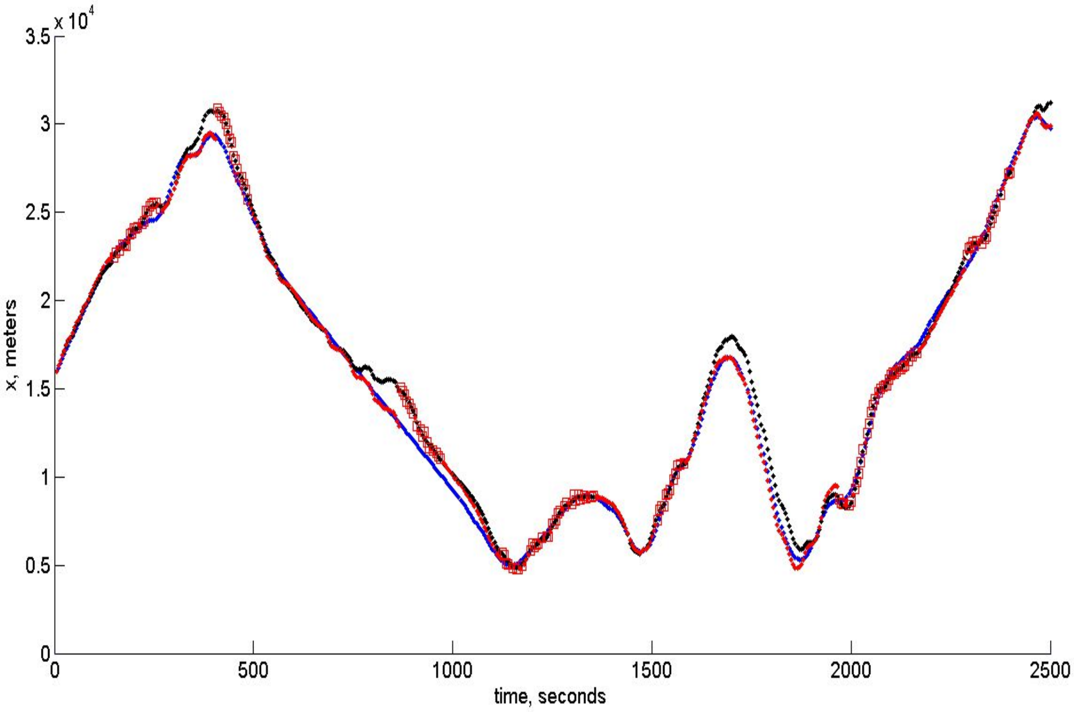

Figure 8. Tracking of the *y*-coordinate.

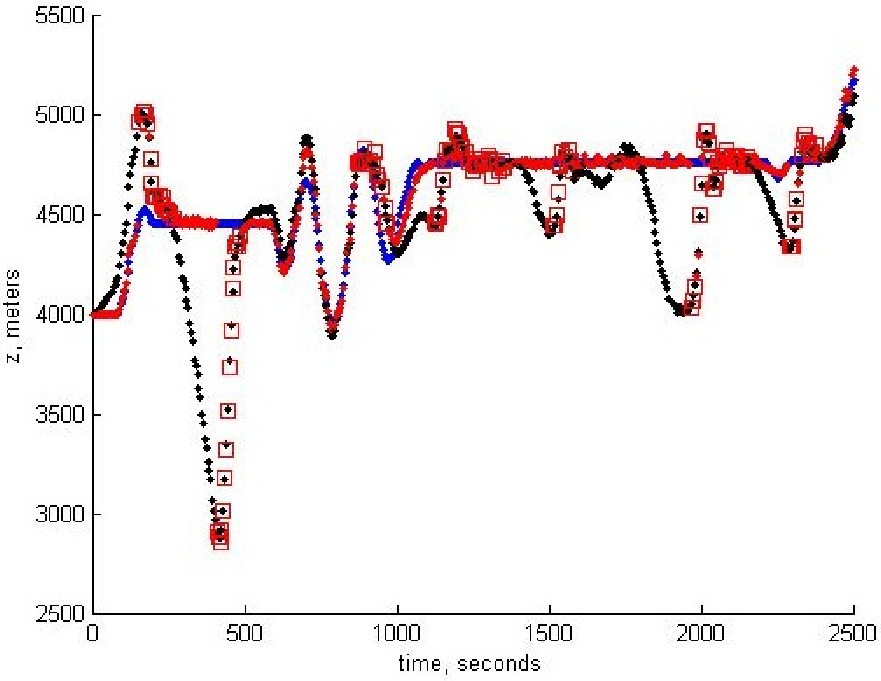

Figure 9. Tracking of the *z*-coordinate.

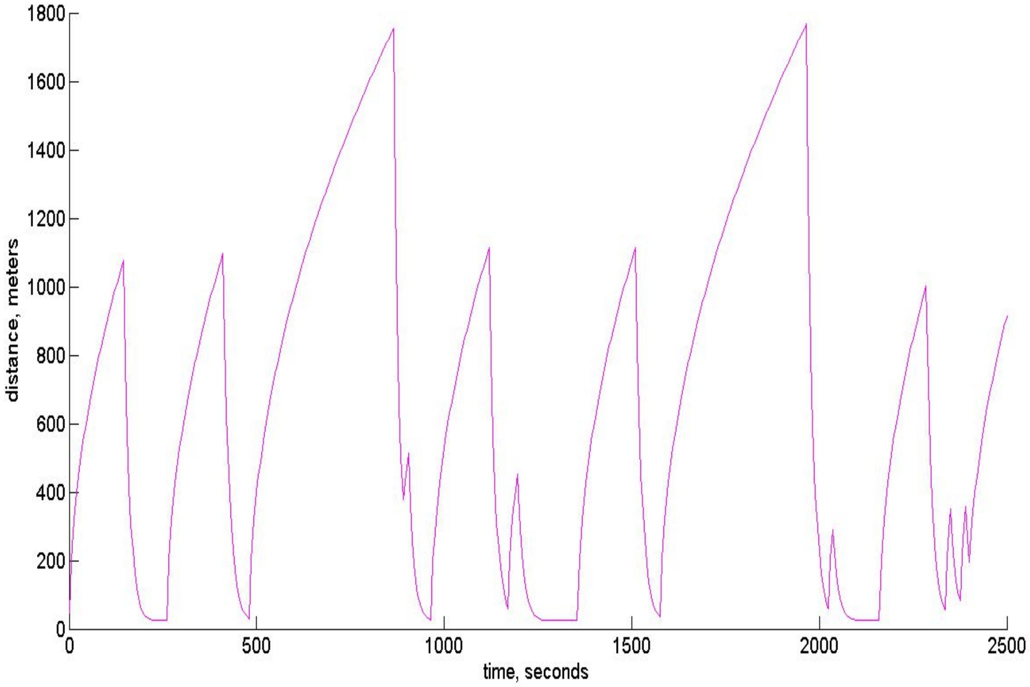

Figure 10. Averaged standard deviation of the position estimation error. The limit value in the observation areas is close to 24.5 m. One can see that in the areas of no observations, the SD monotonically increases.

The high accuracy is in accordance with the standard deviation (SD) theoretically calculated from the PKF. The value of the averaged standard deviation, which is the square root of $P_{xx}(t) + P_{yy}(t) + P_{zz}(t)$, is shown in Figure 10 below.

9. Conclusions

The main result of the work is the new algorithm of the UAV control based on the observation of the landmarks in a 3D environment. The new RANSAC based on the UAV motion model permits one to exclude the huge number of outliers and, by that, to provide the reliable set of data for the estimation of the UAV position on the basis of the novel non-biased PKF algorithm. This work is just the beginning of the implementation of this approach in the navigation of UAVs during long-term autonomous missions.

Acknowledgments: This research was supported by Russian Science Foundation Grant 14-50-00150.

Author Contributions: The work presented here was carried out in collaboration among all authors. All authors have contributed to, seen and approved the manuscript.

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

References

1. Osborn, R.W.; Bar-Shalom, Y. Statistical Efficiency of Composite Position Measurements from Passive Sensors. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2799–2806.
2. Cesetti, A.; Frontoni, E.; Mancini, A.; Zingaretti, P.; Longhi, S. A Vision-Based Guidance System for UAV Navigation and Safe Landing Using Natural Landmarks. *J. Intell. Robot. Syst.* **2010**, *57*, 233–257.
3. Caballero, F.; Merino, L.; Ferruz, J.; Ollero, A. Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *J. Intell. Robot. Syst.* **2009**, *54*, 137–161.
4. Wang, C.L.; Wang, T.M.; Liang, J.H.; Zhang, Y.C.; Zhou, Y. Bearing-only Visual SLAM for Small Unmanned Aerial Vehicles in GPS-denied Environments. *Int. J. Autom. Comput.* **2013**, *10*, 387–396.
5. Konovalenko, I.; Kuznetsova, E. Experimental comparison of methods for estimation the observed velocity of the vehicle in video stream. In Proceedings of the SPIE 9445, Seventh International Conference on Machine Vision (ICMV 2014), Milan, Italy, 19–21 November 2014; Volume 9445.
6. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
7. Konovalenko, I.; Miller, A.; Miller, B.; Nikolaev, D. UAV navigation on the basis of the feature points detection on underlying surface. In Proceedings of the 29th European Conference on Modeling and Simulation (ECMS 2015), Albena (Varna), Bulgaria, 26–29 May 2015; pp. 499–505.
8. Guan, X.; Bai, H. A GPU accelerated real-time self-contained visual navigation system for UAVs. In Proceedings of the IEEE International Conference on Information and Automation, Shenyang, China, 6–8 June 2012; pp. 578–581.
9. Belfadel, D.; Osborne, R.W.; Bar-Shalom, Y. Bias Estimation for Optical Sensor Measurements with Targets of Opportunity. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July, 2013; pp. 1805–1812.
10. Bishop, A.N.; Fidan, B.; Anderson, B.D.O.; Dogancay, K.; Pathirana, P.N. Optimality analysis of sensor-target localization geometries. *Automatica* **2010**, *46*, 479–492.
11. Jauffet, C.; Pillon, D.; Pignoll, A.C. Leg-by-Leg Bearings-only Target Motion Analysis without Observer Maneuver. *J. Adv. Inf. Fusion* **2011**, *6*, 24–38.
12. Lin, X.; Kirubarajan, T.; Bar-Shalom, Y.; Maskell, S. Comparison of EKF, Pseudomeasurement and Particle Filters for a Bearing-only Target Tracking Problem. *Proc. SPIE Int. Soc. Opt. Eng.* **2002**, *4728*, 240–250.
13. Miller, B.M.; Stepanyan, K.V.; Miller, A.B.; Andreev, K.V.; Khoroshenkikh, S.N. Optimal filter selection for UAV trajectory control problems. In Proceeedings of the 37th Conference on Information Technology and Systems, Kaliningrad, Russia, 1–6 September 2013; pp. 327–333.
14. Aidala, V.J.; Nardone, S.C. Biased Estimation Properties of the Pseudolinear Tracking Filter. *IEEE Trans. Aerosp. Electron. Syst.* **1982**, *18*, 432–441.
15. Pugachev, V.S.; Sinitsyn, I.N. *Stochastic Differential Systems—Analysis and Filtering*; Wiley: Chichester, UK, 1987.
16. Amelin, K.S.; Miller, A.B. An Algorithm for Refinement of the Position of a Light UAV on the Basis of Kalman Filtering of Bearing Measurements. *J. Commun. Technol. Electron.* **2014**, *59*, 622–631.

17. Miller, A.B. Development of the motion control on the basis of Kalman filtering of bearing-only measurements. *Autom. Remote Control* **2015**, *76*, 1018–1035.
18. Karpenko, S.; Konovalenko, I.; Miller, A.; Miller, B.; Nikolaev, D. Stochastic control of UAV on the basis of 3D natural landmarks. In Proceedings of the 17th International Conference on Machine Vision (ICMV), Barselona, Spain, 19–20 November 2015.
19. Miller, A.B.; Miller, B.M. Stochastic control of light UAV at landing with the aid of bearing-only observations. In Proceedings of the 17th International Conference on Machine Vision (ICMV), Barselona, Spain, 19–20 November 2015.
20. GitHub. Available online: https://github.com/Itseez/opencv/blob/master/samples/python2/ asift.py (accessed on 23 September 2014).
21. 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.
22. Zuliani, M.; Kenney, C.S.; Manjunath, B.S. The MultiRANSAC Algorithm and Its Application to Detect Planar Homographies. In Proceedings of the 12th IEEE International Conference on Image Processing (ICIP 2005), Genova, Italy, 11–14 September 2005; Volume 3, pp. III-153–156.
23. Civera, J.; Grasa, O.G.; Davison, A.J.; Montiel, J.M.M. 1-Point RANSAC for Extended Kalman Filtering: Application to Real-Time Structure from Motion and Visual Odometry. *J. Field Robot.* **2010**, *27*, 609–631.
24. Miller, B.M.; Pankov, A.R. *Theory of Random Processes (in Russian)*; Phizmatlit: Moscow, Russia, 2007.
25. Torr, P.H.S. Geometric motion segmentation and model selection. *Phil. Trans. R. Soc. A* **1998**, *356*, 1321–1340.
26. Torr, P.H.S.; Zisserman, A. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Comput. Vis. Image Underst.* **2000**, *78*, 138–156.
27. Civera, J.; Grasa, O.G.; Davison, A.J.; Montiel, J.M.M. 1-Point RANSAC for EKF-Based Structure from Motion. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 11–15 October 2009; pp. 3498–3504.
28. Sujit, P.B.; Saripalli, S.; Borges Sousa, J. Unmanned aerial vehicle path following: A survey and analysis of algorithms for Fixed-Wing unmanned aerial vehicless. *IEEE Contol Syst.* **2014**, *34*, 42–59.