

Correction of inertial navigation system's errors by the help of video-based navigator based on Digital Terrain Map.

Oleg Kupervasser, Vladimir Voronov
Transas group of companies
Moscow, Russia
Email: olegkup@yahoo.com

Abstract

This paper deals with the error analysis of a novel navigation algorithm that uses as input the sequence of images acquired from a moving camera and a Digital Terrain (or Elevation) Map (DTM/DEM). More specifically, it has been shown that the optical flow derived from two consecutive camera frames can be used in combination with a DTM to estimate the position, orientation and ego-motion parameters of the moving camera. As opposed to previous works, the proposed approach does not require an intermediate explicit reconstruction of the 3D world. In the present work the sensitivity of the algorithm outlined above is studied. The main sources for errors are identified to be the optical-flow evaluation and computation, the quality of the information about the terrain, the structure of the observed terrain and the trajectory of the camera. By assuming appropriate characterization of these error sources, a closed form expression for the uncertainty of the pose and motion of the camera is first developed and then the influence of these factors is confirmed using extensive numerical simulations. The main conclusion of this paper is to establish that the proposed navigation algorithm generates accurate estimates for reasonable scenarios and error sources, and thus can be effectively used as part of a navigation system of autonomous vehicles.

1 Introduction

Vision-based algorithms has been a major research issue during the past decades. Two common approaches for the navigation problem are: *landmarks* and *ego-motion integration*. In the landmarks approach several features are located on the image-plane and matched to their known 3D location. Using the 2D and 3D data the camera's pose can be derived. Few examples for such algorithms are [2], [3]. Once the landmarks were found, the pose derivation is simple and can achieve quite accurate estimates. The main difficulty is the detection of the features and their correct matching to the landmarks set.

In ego-motion integration approach the motion of the camera with respect to itself is estimated. The ego-motion can be derived from the optical-flow field, or from instruments such as accelerometers and gyroscopes. Once the ego-motion was obtained, one can integrate this motion to derive the camera's path. One of the factors that make this approach attractive is that no specific features need to be detected, unlike the previous approach. Several ego-motion estimation algorithms can be found in [4], [5], [6], [7]. The weakness of ego-motion integration comes from the fact that small errors are accumulated during the integration process. Hence, the estimated camera's path is drifted and the pose estimation accuracy decrease along time. If such approach is used it would be desirable to reduce the drift by activating, once in a while, an additional algorithm that estimates the pose directly. In [8], such navigation-system is being suggested. In that work, like in this work, the drift is being corrected using a Digital Terrain Map (DTM). The DTM is a discrete representation of the observed ground's topography. It contains the altitude over the sea level of the terrain for each geographical location. In [8] a patch from the ground was reconstructed using 'structure-from-motion' (SFM) algorithm and was matched to the DTM in order to derive the camera's pose. Using SFM algorithm which does not make any use of the information obtained from the DTM but rather bases its estimate on the flow-field alone, positions their technique under the same critique that applies for SFM algorithms [1].

The algorithm presented in this work does not require an intermediate explicit reconstruction of the 3D world. By combining the DTM information directly with the images information it is claimed that the algorithm is well-conditioned and

generates accurate estimates for reasonable scenarios and error sources. In the present work this claim is explored by performing an error analysis on the algorithm outlined above. By assuming appropriate characterization of these error sources, a closed form expression for the uncertainty of the pose and motion of the camera is first developed and then the influence of different factors is studied using extensive numerical simulations.

2. Problem Definition and Notations

The problem can be briefly described as follows: At any given time instance t , a coordinates system $C(t)$ is fixed to a camera in such a way that the Z -axis coincides with the optical-axis and the origin coincides with the camera's projection center. At that time instance the camera is located at some geographical location $p(t)$ and has a given orientation $R(t)$ with respect to a global coordinates system W ($p(t)$ is a 3D vector, $R(t)$ is an orthonormal rotation matrix). $p(t)$ and $R(t)$ define the transformation from the camera's frame $C(t)$ to the world's frame W , where if ${}^c v$ and ${}^w v$ are vectors in $C(t)$ and W respectively, then ${}^w v = R(t) {}^c v + p(t)$.

Consider now two sequential time instances t_1 and t_2 : the transformation from $C(t_1)$ to $C(t_2)$ is given by the translation vector $\Delta p(t_1, t_2)$ and the rotation matrix $\Delta R(t_1, t_2)$, such that ${}^{c(t_2)} v = \Delta R(t_1, t_2) {}^{c(t_1)} v + \Delta p(t_1, t_2)$. A rough estimate of the camera's pose at t_1 and of the ego-motion between the two time instances - $p_E(t_1), R_E(t_1), \Delta p_E(t_1, t_2)$ and $\Delta R_E(t_1, t_2)$ - are supplied (the subscript letter "E" denotes that this is an estimated quantity).

Also supplied is the optical-flow field: $\{u_i(t_k)\}$ ($i=1 \dots n, k=1,2$). For the i 'th feature, $u_i(t_1) \in \mathbb{R}^2$ and $u_i(t_2) \in \mathbb{R}^2$ represent its locations at the first and second frame respectively.

Using the above notations, the objective of the proposed algorithm is to estimate the true camera's pose and ego-motion: $p(t_1), R(t_1), \Delta p(t_1, t_2)$ and $\Delta R(t_1, t_2)$, using the optical-flow field $\{u_i(t_k)\}$, the DTM and the initial-guess: $p_E(t_1), R_E(t_1), \Delta p_E(t_1, t_2)$ and $\Delta R_E(t_1, t_2)$.

3. The Navigation Algorithm

The following section describes a navigation algorithm which estimate the above mentioned parameters. The pose and ego-motion of the camera are derived using a DTM and the optical-flow field of two consecutive frames. Unlike the landmarks approach no specific features should be detected and matched. Only the correspondence between the two consecutive images should be found in order to derive the optical-flow field. As was mentioned in the previous section, a rough estimate of the required parameters is supplied as an input. Nevertheless, since the algorithm only use this input as an initial guess and re-calculate the pose and ego-motion directly, no integration of previous errors will take place and accuracy will be preserved.

The new approach is founded on the following observation. Since the DTM supplies information about the structure of the observed terrain, depth of observed features is being dictated by the camera's pose. Hence, given the pose and ego-motion of the camera, the optical-flow field can be uniquely determined. The objective of the algorithm will be finding the pose and ego-motion which lead to an optical-flow field as close as possible to the given flow field.

A single vector from the optical-flow field will be used to define a constraint for the camera's pose and ego-motion. Let ${}^w G \in \mathbb{R}^3$ be a location of a ground feature point in the 3D world. At two different time instances t_1 and t_2 , this feature point is projected on the image-plane of the camera to the points $u(t_1)$ and $u(t_2)$. Assuming a pinhole model for the camera, then $u(t_1), u(t_2) \in \mathbb{R}^2$. Let ${}^c q(t_1)$ and ${}^c q(t_2)$ be the homogeneous representations of these locations. As standard, one can think of these vectors as the vectors from the optical-center of the camera to the projection point on the image plane. Using an initial-guess of the pose of the camera at t_1 , the line passing through $p_E(t_1)$ and ${}^c q(t_1)$ can be intersected with the DTM. Any ray-tracing style algorithm can be used for this purpose. The location of this intersection is denoted as ${}^w G_E$. The subscript letter "E" highlights the fact that this ground-point is the estimated location for the feature point, that in general will be different from the true ground-feature location ${}^w G$. The difference between the true and estimated locations is due to two main sources: the error in the initial guess for the pose and the errors in the determination of ${}^w G_E$ caused by DTM discretization and intrinsic errors. For a reasonable initial-guess and DTM-related errors, the two points ${}^w G_E$ and ${}^w G$ will be close enough so as to allow the linearization of the DTM around ${}^w G_E$. Denoting by N the normal of the plane tangent to the DTM at the point ${}^w G_E$, one can write:

$$N^T ({}^w G - {}^w G_E) \approx 0 \quad (1)$$

The true ground feature ${}^w G$ can be described using true pose parameters:

$${}^w G = R(t_1) \cdot {}^c q(t_1) \cdot \lambda + p(t_1) \quad (2)$$

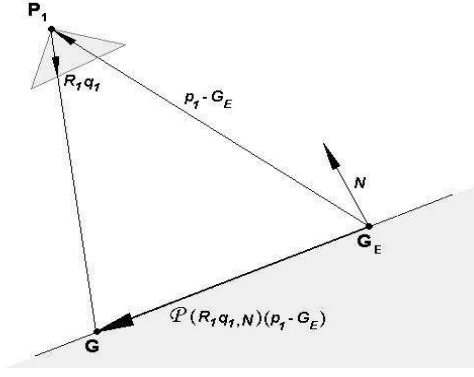


Figure 1: Geometrical description of expression (9) using the projection operator (7)

Here, λ denotes the depth of the feature point (i.e. the distance of the point to the image plane projected on the optical-axis). Replacing (2) in (1):

$$N^T(\lambda \cdot R(t_1) \cdot {}^c q(t_1) + p(t_1) - {}^w G_E) = 0 \quad (3)$$

From this expression, the depth of the true feature can be computed using the estimated feature location:

$$\lambda = \frac{N^T {}^w G_E - N^T p(t_1)}{N^T R(t_1) {}^c q(t_1)} \quad (4)$$

By plugging (4) back into (2) one gets:

$${}^w G = R(t_1) {}^c q(t_1) \cdot \left(\frac{N^T {}^w G_E - N^T p(t_1)}{N^T R(t_1) {}^c q(t_1)} \right) + p(t_1) \quad (5)$$

In order to simplify notations, $R(t_i)$ will be replaced by R_i and likewise for $p(t_i)$ and $q(t_i)$ $i = 1, 2$. $\Delta R(t_1, t_2)$ and $\Delta p(t_1, t_2)$ will be replaced by R_{12} and p_{12} respectively. The superscript describing the coordinate frame in which the vector is given will also be omitted, except for the cases where special attention needs to be drawn to the frames. Normally, p_{12} and q 's are in camera's frame while the rest of the vectors are given in the world's frame. Using the simplified notations, (5) can be rewritten as:

$$G = \frac{R_1 q_1 N^T}{N^T R_1 q_1} G_E - \frac{R_1 q_1 N^T}{N^T R_1 q_1} p_1 + p_1 \quad (6)$$

In order to obtain simpler expressions, define the following projection operator:

$$\mathcal{P}(u, s) \doteq \left(\mathbf{I} - \frac{us^T}{s^T u} \right) \quad (7)$$

This operator projects a vector onto the subspace normal to s , along the direction of u . As an illustration, it is easy to verify that $s^T \cdot \mathcal{P}(u, s)v \equiv 0$ and $\mathcal{P}(u, s)u \equiv 0$. By adding and subtracting G_E to (6), and after reordering:

$$G = G_E + \left[\mathbf{I} - \frac{R_1 q_1 N^T}{N^T R_1 q_1} \right] p_1 - \left[\mathbf{I} - \frac{R_1 q_1 N^T}{N^T R_1 q_1} \right] G_E \quad (8)$$

Using the projection operator, (8) becomes:

$$G = G_E + \mathcal{P}(R_1 q_1, N) (p_1 - G_E) \quad (9)$$

The above expression has a clear geometric interpretation (see Fig.1). The vector from G_E to p_1 is being projected onto the tangent plane. The projection is along the direction $R_1 q_1$, which is the direction of the ray from the camera's optical-center (p_1), passing through the image feature.

Our next step will be transferring G from the global coordinates frame- W into the first camera's frame C_1 and then to the second camera's frame C_2 . Since p_1 and R_1 describe the transformation from C_1 into W , we will use the inverse transformation:

$${}^{c_2}G = p_{12} + R_{12} (R_1^T (G - p_1)) \quad (10)$$

Assigning (9) into (10) gives:

$${}^{c_2}G = p_{12} + R_{12} \mathcal{L} (G_E - p_1) \quad (11)$$

\mathcal{L} in the above expression represents:

$$\mathcal{L} = \frac{q_1 N^T}{N^T R_1 q_1} \quad (12)$$

One can think of \mathcal{L} as an operator with inverse characteristic to \mathcal{P} : it projects vectors on the ray continuing $R_1 q_1$ along the plane orthogonal to N .

q_2 is the projection of the true ground-feature G . Thus, the vectors q_2 and ${}^{c_2}G$ should coincide. This observation can be expressed mathematically by projecting ${}^{c_2}G$ on the ray continuation of q_2 :

$${}^{c_2}G = \frac{q_2}{|q_2|} \cdot \left(\frac{q_2^T}{|q_2|} \cdot {}^{c_2}G \right) \quad (13)$$

In expression (13), $q_2^T / |q_2| \cdot {}^{c_2}G$ is the magnitude of ${}^{c_2}G$'s projection on q_2 . By reorganizing (13) and using the projection operator, we obtain:

$$\left[\mathbf{I} - \frac{q_2 \cdot q_2^T}{q_2^T \cdot q_2} \right] \cdot {}^{c_2}G = \mathcal{P}(q_2, q_2) \cdot {}^{c_2}G = 0 \quad (14)$$

${}^{c_2}G$ is being projected on the orthogonal complement of q_2 . Since ${}^{c_2}G$ and q_2 should coincide, this projection should yield the zero-vector. Plugging (11) into (14) yields our final constraint:

$$\mathcal{P}(q_2, q_2) [p_{12} + R_{12} \mathcal{L} (G_E - p_1)] = 0 \quad (15)$$

This constraint involves the position, orientation and the ego-motion defining the two frames of the camera. Although it involves 3D vectors, it is clear that its rank can not exceed two due to the usage of \mathcal{P} which projects \mathbb{R}^3 on a two-dimensional subspace.

Such constraint can be established for each vector in the optical-flow field, until a non-singular system is obtained. Since twelve parameters need to be estimated (six for pose and six for the ego-motion), at least six optical-flow vectors are required for the system solution. But it is correct conclusion for nonlinear problem. If we use Gauss-Newton iterations method and so make linearization of our problem near approximate solution. The found matrix will be always singular for six points (with zero determinant) as numerical simulations demonstrate. So it is necessary to use at least seven points to obtain nonsingular linear approximation. Usually, more vectors will be used in order to define an over-determined system, which will lead to more robust solution. The reader attention is drawn to the fact that a non-linear constraint was obtained. Thus, an iterative scheme will be used in order to solve this system. A robust algorithm which uses Gauss-Newton iterations and M-estimator is described in [9]. We begin to use Levenberg-Marquardt method if Gauss-Newton method after several iterations stopped to converge. This two algorithms are realized in `lsqnonlin()` Matlab function. The applicability, accuracy and robustness of the algorithm was verified though simulations and lab-experiments.

It is more convenient to use more robust for iterations equivalent to (15) equation:

$$\mathcal{P}(q_2, q_2) [p_{12} + R_{12} \mathcal{L}_i (G_{E_i} - p_1)] / |{}^{c_2}G| = 0 \quad (16)$$

Using of this normalized form of equations avoids to get incorrect trivial solution when two positions are in a single point on the ground.

3.1 Multiple Features

Suppose next that n feature points are tracked in two frames, so that the estimated locations Q_{E_i} and projections onto the image plane q_{1i} and q_{2i} are estimated and measured, respectively, for $i = 1, \dots, n$. Associated with each Q_{E_i} is the normal vector to the DTM at this point, namely N_i .

Taking this into account, one can re-write (15) in matrix form as:

$$\begin{bmatrix} -\mathcal{P}(q_{2i}) & \mathcal{P}(q_{2i}) \frac{R_{12}q_{1i}N_i^T}{N_i^T R_1 q_{1i}} \end{bmatrix} \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = \mathcal{P}(q_{2i}) \frac{R_{12}q_{1i}N_i^T}{N_i^T R_1 q_{1i}} Q_{Ei}. \quad (17)$$

Repeating this for each feature point:

$$\begin{bmatrix} -\mathcal{P}(q_{21}) & \mathcal{P}(q_{21}) \frac{R_{12}q_{11}N_1^T}{N_1^T R_1 q_{11}} \\ -\mathcal{P}(q_{22}) & \mathcal{P}(q_{22}) \frac{R_{12}q_{12}N_2^T}{N_2^T R_1 q_{12}} \\ \vdots & \vdots \\ -\mathcal{P}(q_{2n}) & \mathcal{P}(q_{2n}) \frac{R_{12}q_{1n}N_n^T}{N_n^T R_1 q_{1n}} \end{bmatrix} \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = \begin{bmatrix} \mathcal{P}(q_{21}) \frac{R_{12}q_{11}N_1^T}{N_1^T R_1 q_{11}} Q_{E1} \\ \mathcal{P}(q_{22}) \frac{R_{12}q_{12}N_2^T}{N_2^T R_1 q_{12}} Q_{E2} \\ \vdots \\ \mathcal{P}(q_{2n}) \frac{R_{12}q_{1n}N_n^T}{N_n^T R_1 q_{1n}} Q_{En} \end{bmatrix} \quad (18)$$

In compact notation:

$$\mathcal{A}_n \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = \mathcal{B}_n. \quad (19)$$

Note that \mathcal{A}_n and \mathcal{B}_n depend on known quantities: the estimated features, the normals of the DTM tangent planes, and the images of the features at the two time instances, together with the unknown orientation R_1 and the relative rotation R_{12} . At this point in our discussion, several remarks are in order.

Remark 1: The constraint (18) involves twelve "unknowns", namely the pose and ego-motion of the camera. From the remark at the end of the previous section, the equation involves at most $2n$ linearly independent constraints, so that at least six features at different locations Q_{Ti} are required to have a determinate system of equations. But it is correct conclusion for nonlinear problem. If we use Gauss-Newton iterations method and so make linearization of our problem near approximate solution. The found matrix will be always singular for six points (with zero determinant) as numerical simulations demonstrate. So it is necessary to use at least seven points to obtain nonsingular linear approximation. Usually, more vectors will be used in order to define an over-determined system, and hence reduce the effect of noise. Clearly, there are degenerate scenarios in which the obtained system is singular, no matter what is the number of available features. Examples for such scenarios include flying above completely planar or spherical terrain. However, in the general case where the terrain has "interesting" structure the system is non-singular and the twelve parameters can be obtained.

Remark 2: The constraint (18) is non-linear and, therefore, no analytic solution to it is readily available. Thus, an iterative scheme will be used in order to solve this system. A robust algorithm using Newton-iterations and M-estimator will be described in following sections.

Remark 3: Given Remark 2, one observes that the location and translation appear linearly in the constraint. Using the pseudo-inverse, these two vectors can be solved explicitly to give:

$$\begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = \mathcal{A}_n^\dagger \mathcal{B}_n, \quad (20)$$

so that, after resubstituting in (19):

$$(I - \mathcal{A}_n \mathcal{A}_n^\dagger) \mathcal{B}_n = 0. \quad (21)$$

This remark leads to two conclusions:

1. If the rotation is known to good accuracy and measurement noise is relatively low, then the position and translation can be determined by solving a linear equation. This fact may be relevant when "fusing" the procedure described here with other measurement, e.g., with inertial navigation.

2. Equation (21) shows that the estimation of rotation (both absolute and relative) can be separated from that of location/translation. This fact is also found when estimating pose from a set of visible landmarks as shown in [17]. In that work, similarly to the present, the estimate is obtained by minimizing an objective function which measures the errors in the *object-space* rather than on the image plane (as in most other works). This property enables the decoupling of the estimation problem. Note however that [17] address's only the pose rotation and translation decoupling while here the 6 parameters of absolute and relative rotations are separated from the 6 parameters of the camera location and translation.

3.2 The Epipolar Constraint Connection

Before proceeding any further, it is interesting to look at (15) in the light of previous work in SFM and, in particular, epipolar geometry. In order to do this, it is worth deriving the basic constraint in the present framework and notation. Write:

$${}^{c_2}Q_T = \lambda_2 q_2 = p_{12} + \lambda_1 R_{12} q_1 \quad (22)$$

for some scalars λ_1 and λ_2 (see Fig.2).

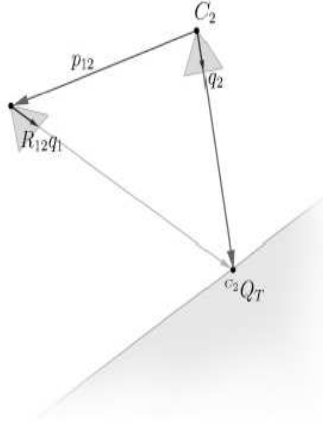


Figure 2: The examined scenario from the second camera frame's (C_2) point of view. q_2 is the perspective projection of the terrain feature ${}^{c_2}Q_T$, and thus the two should coincide. Additionally, since q_1 is also a projection of the same feature in the C_1 -frame, the epipolar constraint requires that the two rays (one in the direction of q_2 and the other from p_{12} in the direction of $R_{12}q_1$) will intersect.

It follows that:

$$p_{12} \times \lambda_2 q_2 = p_{12} \times \lambda_1 R_{12} q_1, \quad (23)$$

and hence:

$$q_2^T (p_{12} \times R_{12} q_1) = 0. \quad (24)$$

For a vector $x \in \mathbb{R}^3$, let x^\wedge denote the skew-symmetric matrix:

$$x^\wedge = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Then, it is well known that the vector product between two vectors x and y can be expressed as:

$$x \times y = x^\wedge y.$$

Using this notation, the epipolar constraint (24) can be written as:

$$q_2^T (R_{12}q_1)^\wedge p_{12} = 0 \quad (25)$$

and symmetrically as:

$$q_1^T R_{12}^T q_2^\wedge p_{12} = 0 \quad (26)$$

The important observation here is that if the vector p_{12} verifies the above constraint, then the vector $\kappa \cdot p_{12}$ also verifies the constraint, for any number κ . This is an expression of the ambiguity built into the SFM problem. On the other hand, the constraint (15) is non-homogeneous and hence does not suffer from the same ambiguity. In terms of the translation alone (and for only one feature point!), if p_{12} verifies (15) for given R_1 and R_{12} , then also $p_{12} + \kappa q_2$ will verify the constraint, and hence the ego-motion translation is defined up to a one-dimensional vector. However, one has the following trivially:

$$q_1^T R_{12}^T q_2^\wedge q_2 = 0, \quad (27)$$

and hence the epipolar constraint does not provide an additional equation that would allow us to solve for the translation in a unique manner. Moreover, observe that (15) can be written using a vector product instead of the projection operator as:

$$q_2^\wedge \left[p_{12} + \frac{R_{12}q_1 N^T}{N^T R_1 q_1} (Q_E - p_1) \right] = 0. \quad (28)$$

Taking into account the identity

$$(R_{12}q_1)^T q_2^\wedge R_{12}q_1 \equiv 0, \quad (29)$$

it is possible to conclude that (28) \longrightarrow (26), and hence the new constraint "contains" the classical epipolar geometry. Indeed, one could think of the constraint derived in (15) as strengthening the epipolar constraint by requiring not only that the two rays (in the directions of q_1 and q_2) should intersect, but, in addition, that this intersection point should lie on the DTM's linearization plane. Observe, moreover, that taking more than one feature point would allow us to completely compute the translation (at least for the given rotation matrices).

4 Vision-based navigation algorithm corrections for inertial navigation by help of Kalman filter.

Vision-based navigation algorithms has been a major research issue during the past decades. Algorithm used in this paper is based on foundations of multiple-view geometry and a land map. By help of this method we get position and orientation of a observer camera. On the other hand we obtain the same data from inertial navigation methods. To adjust these two results Kalman filter is used. We employ in this paper extended Kalman filter for nonlinear equations [12].

For inertial navigation computations was used Inertial Navigation System Toolbox for Matlab [13].

Input of Kalman filter consists of two part. The first one is variables X for equations of motion. In our case it is inertial navigation equations. Vector X consists of fifteen components: $[\delta x \ \delta y \ \delta z \ \delta V_x \ \delta V_y \ \delta V_z \ \delta \phi \ \delta \theta \ \delta \psi \ a_x \ a_y \ a_z \ b_x \ b_y \ b_z]$. Coordinates $\delta x \ \delta y \ \delta z$ are defined by difference between real position of the camera and position gotten from inertial navigation calculus. Variables $\delta V_x \ \delta V_y \ \delta V_z$ are defined by difference between real velocity of the camera and velocity gotten from inertial navigation calculus. Variable $\delta \phi \ \delta \theta \ \delta \psi$ are defined as Euler angles of matrix $D_r * D_c^T$ where D_r is matrix defined by real Euler angles of camera with respect to Local Level Frame (L-Frame) and D_c is matrix defined by Euler angles of camera with respect to Local Level Frame (L-Frame) gotten by inertial navigation computation. It is necessary to pay attention that found Euler angles $\delta \phi \ \delta \theta \ \delta \psi$ ARE NOT equivalent to difference between real Euler angles and Euler angles gotten from inertial navigation calculus. For small values of $\delta \phi \ \delta \theta \ \delta \psi$ perturbations to these angles can be added linearly and so these angles can be used in Kalman filter for small errors. Such choose of angles is made because formulas describing their evolution are much simpler than formulas describing evolution of Euler angles differences. Variables $a_x \ a_y \ a_z$ are defined by vector of Accel bias in inertial navigation measurements. Variables $b_x \ b_y \ b_z$ are defined by vector of Gyro bias in inertial navigation measurements.

The second input of Kalman filter is Z -result of measurements by vision-based navigation algorithms. Vector Z consists of six components $[\delta x_m \ \delta y_m \ \delta z_m \ \delta \phi_m \ \delta \theta_m \ \delta \psi_m]$ Coordinates $\delta x_m \ \delta y_m \ \delta z_m$ are difference between camera position measured by vision-based navigation algorithm and position gotten from inertial navigation calculus. Variable $\delta \theta_m \ \delta \psi_m$ are defined as Euler angles of matrix $D_m * D_c^T$ where D_m is matrix defined by Euler angles of camera with respect to Local Level Frame

(L-Frame) measured by vision-based navigation algorithm and D_c is matrix defined by Euler angles of camera with respect to Local Level Frame (L-Frame) gotten by inertial navigation computation. Let variable k to be number of step for time discretization used in Kalman filter.

We assume that errors for between values gotten by inertial navigation computation and real values are linearly depend on noise. Correspondent process noise covariance matrix is denoted by Q_k . Diagonal elements of Q_k correspondent to velocity are defined by Accel noise and proportional to dt^2 : $Q_V \sim dt^2$, where dt is time interval between t_k and t_{k-1} : $dt = t_k - t_{k-1}$. Diagonal elements of Q_k correspondent to Euler angles are defined by Gyro noise and proportional to dt : $Q_A \sim dt$.

We assume that errors for between values gotten by vision-based navigation algorithm and real values are linearly depend on noise. Correspondent measurement noise covariance matrix is denoted by R_k . Error analysis giving this matrix is described in [14].

Kalman filter equations describe evolution of *a posteriori* state estimation X_k described above and *a posteriori* error covariation covariance matrix P_k for variables X_k .

To write Kalman filter equations we must define two 15x15 matrices yet: H_k and A_k . Matrix H_k is measurement Jacobian describing connection between predicted measurement $H_k * X_k$ and actual measurement Z_k defined above. Diagonal elements $H_k(1, 1)$, $H_k(2, 2)$, $H_k(3, 3)$ describing coordinate and elements $H_k(4, 7)$, $H_k(5, 8)$, $H_k(6, 9)$ describing angles are equal to one. The rest of the elements are equal to zero.

A_k is Jacobian matrix describing evolution of vector X_k . The exact expression for this matrix is very difficult so we use approximate formula for A_k neglecting by Coriolis effects, Earth rotation and so on. Let ϕ θ ψ be the Euler angles in L-Frame, dV is deltaV vector gotten from inertial navigation measurements, f_{vec} is acceleration vector in L-frame, DCM_{b-to-l} is direction cosine matrix (from body-frame to L-frame).

The formulas defining A_k are follow:

$$\Psi_{DCM} = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (30)$$

$$\Theta_{DCM} = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (31)$$

$$\Phi_{DCM} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (32)$$

$$DCM_{b-to-l} = \Phi_{DCM} \Theta_{DCM} \Psi_{DCM} \quad (33)$$

$$f_{vec} = DCM_{b-to-l} \frac{dV}{dt} \quad (34)$$

$$Phi(1 : 3, 4 : 6) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (35)$$

$$Phi(4 : 6, 7 : 9) = \begin{pmatrix} 0 & -f_{vec}(3) & f_{vec}(2) \\ f_{vec}(3) & 0 & -f_{vec}(1) \\ -f_{vec}(2) & f_{vec}(1) & 0 \end{pmatrix} \quad (36)$$

$$Phi(7 : 9, 10 : 12) = -DCM_{b-to-l} \quad (37)$$

$$Phi(4 : 6, 13 : 15) = -DCM_{b-to-l} \quad (38)$$

The rest of elements for matrix Phi are equal to zero.

$$A_k = I + Phi dt \quad (39)$$

Kalman filter time update equations are follow:

$$X_k^- = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ a_{x_{k-1}} \ a_{y_{k-1}} \ a_{z_{k-1}} \ b_{x_{k-1}} \ b_{y_{k-1}} \ b_{z_{k-1}}] \quad (40)$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1} \quad (41)$$

Kalman filter update equations project the state and covariance estimates from the previous time step $k - 1$ to the current time step k .

Kalman filter measurement update equations are follow:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (42)$$

$$X_k = X_k^- + K_k (Z_k - H_k X_k^-) \quad (43)$$

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (44)$$

Kalman filter measurement update equations correct the state and covariance estimates with measurement Z_k .

The found vector X_k is used to update coordinates, velocities, Euler angles, Accel and Gyro biases for inertial navigation calculations on the next step.

Numerical simulations were realized to examine effectiveness of Kalman filter to combine these two navigation algorithms. On figurefig:fig1 we can see that corrected path for coordinate error much smaller than inertial navigation coordinate error without Kalman filter. Improved results by help Kalman filter are gotten also for velocity in spite of the fact that this velocity was not measured by help vision-based navigation algorithm.

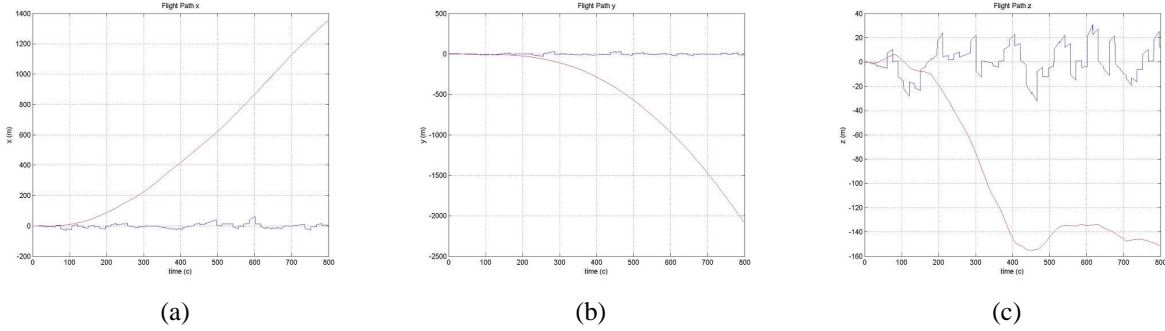


Figure 3: Position errors ((a) for x coordinate (b) for y coordinate (c) for z coordinate) of the drift path are marked with a red line, and errors of the corrected path are marked with a blue line. Parameters : Height 1000m, FOV 60 degree, Features number 120, Resolution 1000x1000, Baseline=200m, $\Delta time = 15$ s

5 Error analysis

The rest of this work deals with the error-analysis of the proposed algorithm. In order to evaluate the algorithm's performance, the objective-function of the minimization process needs to be defined first: For each of the n optical-flow vectors, the function $f_i : \mathbb{R}^{12} \rightarrow \mathbb{R}^3$ is defined as the left-hand side of the constraint described in (16):

$$\begin{aligned} f_i(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) &= \\ &= \mathcal{P}(q_2, q_2) [p_{12} + R_{12} \mathcal{L}_i (G_{E_i} - p_1)] / |{}^C_2 G| \end{aligned} \quad (45)$$

In the above expression, R_{12} and \mathcal{L}_i are functions of $(\phi_{12}, \theta_{12}, \psi_{12})$ and $(\phi_1, \theta_1, \psi_1)$ respectively. Additionally, the function $F : \mathbb{R}^{12} \rightarrow \mathbb{R}^{3n}$ will be defined as the concatenation of the f_i functions: $F(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) = [f_1, \dots, f_n]^T$. According to these notations, the goal of the algorithm is to find the twelve parameters that minimize $M(\theta, D) = \|F(\theta, D)\|^2$, where θ represents the 12-vector of the parameters to be estimated, and D is the concatenation of all the data

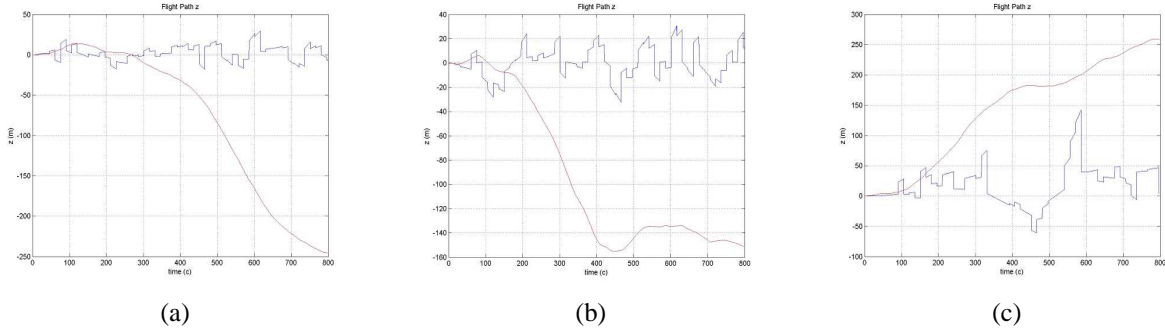


Figure 4: Position errors for x, y, z coordinate of the drift path are marked with a red line, and errors of the corrected path are marked with a blue line. Parameters : FOV 60 degree, Features number 120, Resolution 1000x1000, Baseline=200m, $\Delta time = 15$ s, Height a) 700m b) 1000m c) 3000m

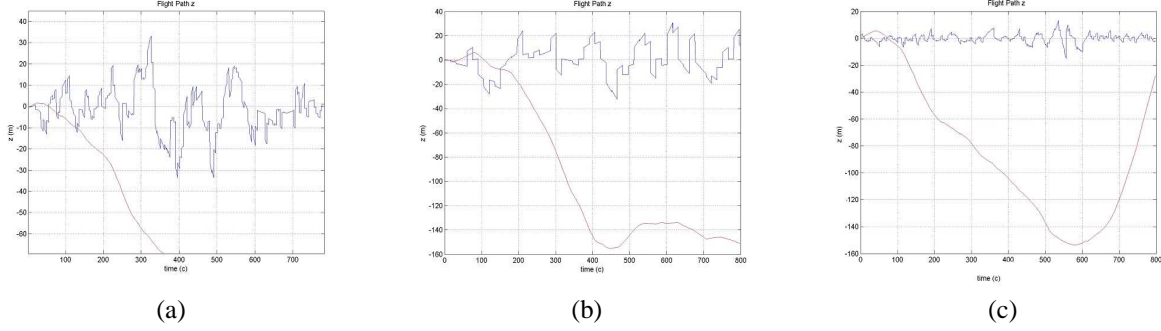


Figure 5: Position errors for x, y, z coordinate of the drift path are marked with a red line, and errors of the corrected path are marked with a blue line. Parameters : FOV 60 degree, Features number 120, Baseline=200m, $\Delta time = 15$ s, Height 1000m, Resolution a) 500x500 b) 1000x1000 c) 4000x4000

obtain from the optical-flow and the DTM. If D would have been free of errors, the true parameters were obtained. Since D contains some error perturbation, the estimated parameters are drifted to erroneous values. It has been shown in [10] that the connection between the uncertainty of the data and the uncertainty of the estimated parameters can be described by the following first-order approximation:

$$\Sigma_{\theta} = \left(\frac{dg}{d\theta} \right)^{-1} \left(\frac{dg}{dD} \right) \Sigma_D \left(\frac{dg}{dD} \right)^T \left(\frac{dg}{d\theta} \right)^{-1} \quad (46)$$

Here, Σ_{θ} and Σ_D represent the covariance matrices of the parameters and the data respectively. g is defined as follows:

$$g(\theta, D) \doteq \frac{d}{d\theta} M(\theta, D) = \frac{d}{d\theta} F^T F = 2J_{\theta}^T F \quad (47)$$

$J_{\theta} = dF/d\theta$ is the $(3n \times 12)$ Jacobian matrix of F with respect to the twelve parameters. By ignoring second-order elements, the derivations of g can be approximate by:

$$\frac{dg}{d\theta} \approx 2J_{\theta}^T J_{\theta} \quad (48)$$

$$\frac{dg}{dD} \approx 2J_{\theta}^T J_D \quad (49)$$

$J_D = dF/dD$ is defined in a similar way as the $(3n \times m)$ Jacobian matrix of F with respect to the m data components. Assigning (48) and (49) back into (46) yield the following expression:

$$J_T = (J_{\theta}^T J_{\theta})^{-1} J_{\theta}^T$$

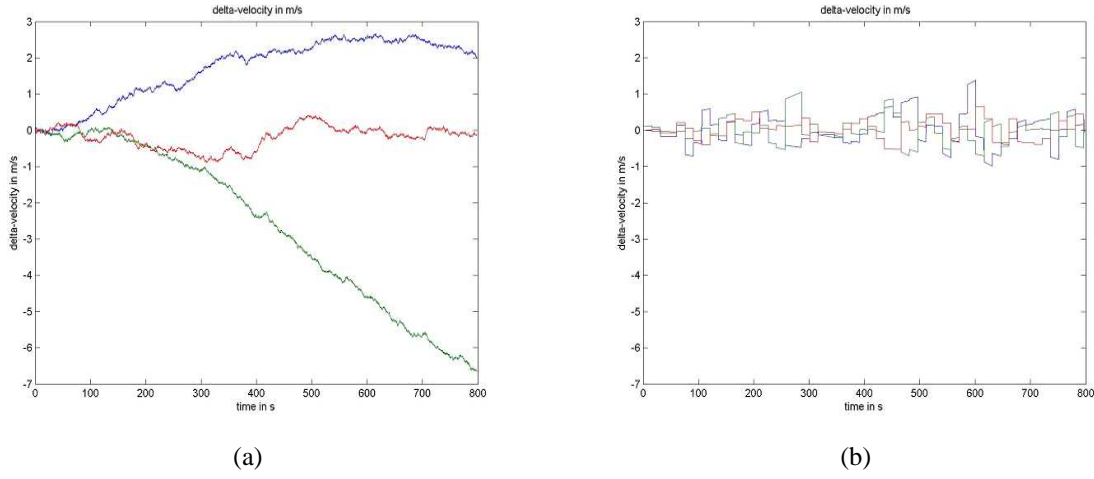


Figure 6: (a) Velocity errors of the drift path (x y z components), and (b) Velocity errors of the corrected path (x y z components). Parameters Height 1000m, FOV 60 degree, Features number 120, Resolution 1000x1000, Baseline=200m, $\Delta time = 15$ s

$$\Sigma_{\theta} = J_T \cdot (J_D \Sigma_D J_D^T) \cdot J_T^T \quad (50)$$

The central component $J_D \Sigma_D J_D^T$ represents the uncertainties of F while the pseudo-inverse matrix $(J_{\theta}^T J_{\theta})^{-1} J_{\theta}^T$ transfers the uncertainties of F to those of the twelve parameters. In the following subsections, J_{θ} , J_D and Σ_D are explicitly derived.

5.1. J_{θ} Calculation

Simple derivations of f_i which is presented in (45), yield the following results:

$$N_P(q_2, {}^{c_2}G) = \mathcal{P}(q_2, q_2) \mathcal{P}({}^{c_2}G, {}^{c_2}G) / |{}^{c_2}G| \quad (51)$$

$$\frac{df}{dp_1} = -N_P(q_2, {}^{c_2}G) R_{12} \mathcal{L} \quad (52)$$

$$\frac{df}{d\alpha_1} = -N_P(q_2, {}^{c_2}G) R_{12} \mathcal{L} \left(\frac{d}{d\alpha_1} R_1 \right) \mathcal{L} (G_E - p_1) \quad (53)$$

$$\frac{df}{dp_{12}} = N_P(q_2, {}^{c_2}G) \quad (54)$$

$$\frac{df}{d\alpha_{12}} = N_P(q_2, {}^{c_2}G) \left(\frac{d}{d\alpha_{12}} R_{12} \right) \mathcal{L} (G_E - p_1) \quad (55)$$

In expressions (53) and (55): $\alpha_1 = \phi_1, \theta_1, \psi_1$ and: $\alpha_{12} = \phi_{12}, \theta_{12}, \psi_{12}$. The Jacobian J_{θ} is obtained by simple concatenation of the above derivations.

5.2 J_D Calculation

Before calculating J_D , the data vector D must be explicitly defined. Two types of data are being used by the proposed navigation algorithm: data obtained from the optical-flow field and data obtained from the DTM. Each flow vector starts at q_1 and ends at q_2 . One can consider q_1 's location as an arbitrary choice of some ground feature projection, while q_2 represent the new projection of the same feature on the second frame. Thus the flow errors are realized through the q_2 vectors.

The DTM errors influence the G_E and N vectors in the constraint equation. As before, the DTM linearization assumption will be used. For simplicity the derived orientation of the terrain's local linearization, as expressed by the normal, will be considered as correct while the height of this plane might be erroneous. The connection between the height error and the

error of G_E will be derived in the next subsection. Resulting from the above, the q_1 's and the N 's can be omitted from the data vector D . It will be defined as the concatenation of all the q_2 's followed by concatenation of the G_E 's.

The i 'th feature's data vectors: q_{2_i} and G_{E_i} appears only in the i 'th feature constraint, thus the obtained Jacobian matrix $J_D = [J_q, J_G]$ is a concatenation of two block diagonal matrices: J_q followed by J_G . The i 'th diagonal block element is the 3×3 matrix df_i/dq_{2_i} and df_i/dG_{E_i} for J_q and J_G respectively:

$$\frac{df}{dq_2} = \frac{-1}{\|q_2\|^2} [(q_2^T \cdot {}^{c_2}G) \mathbf{I} + q_2 \cdot {}^{c_2}G^T] \mathcal{P}(q_2, q_2) / |{}^{c_2}G| \quad (56)$$

$$\frac{df}{dG_E} = N_P(q_2, {}^{c_2}G) R_{12} \mathcal{L} \quad (57)$$

${}^{c_2}G$ in expression (56) is the ground feature G under the second camera frame as defined in (11).

5.3. Σ_D Calculation

As mention above, the data-vector D is constructed from concatenation of all the q_2 's followed by concatenation of the G_E 's. Thus Σ_D should represent the uncertainty of these elements. Since the q_2 's and the G_E 's are obtained from two different and uncorrelated processed the covariance relating them will be zero, which leads to a two block diagonal matrix:

$$\Sigma_D = \begin{bmatrix} \Sigma_q & 0 \\ 0 & \Sigma_G \end{bmatrix} \quad (58)$$

In this work the errors of image locations and DTM height are assumed to be additive zero-mean Gaussian distributed with standard-deviation of σ_I and σ_h respectively. Each q_2 vector is a projection on the image plane where a unit focal-length is assumes. Hence, there is no uncertainty about its z -component. Since a normal isotropic distribution was assumed for the sake of simplicity, the covariance matrix of the image measurements is defined to be:

$$\Sigma_{q_i} = \sigma_I^2 \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \quad (59)$$

and Σ_q is the matrix with the Σ_{q_i} 's along its diagonal.

In [11] the accuracy of location's height obtained by interpolation of the neighboring DTM grid points is studied. The dependence between this accuracy and the specific required location, for which height is being interpolated, was found to be negligible. Here, the above finding was adopted and a constant standard-deviation was set to all DTM heights measurements. Although there is a dependence between close G_E 's uncertainties, this dependence will be ignored in the following derivations for the sake of simplicity. Thus, a block diagonal matrix is obtained for Σ_G containing the 3×3 covariance matrices Σ_{G_i} along its diagonal which will be derived as follows: consider the ray sent from p_1 along the direction of $R_1 q_1$. This ray should have intersected the terrain at $G_E = p_1 + \lambda R_1 q_1$ for some λ , but due to the DTM height error the point $\tilde{G}_E = (\tilde{x}, \tilde{y}, \tilde{h})^T$ was obtained. Let h be the true height of the terrain above (\tilde{x}, \tilde{y}) and $H = (\tilde{x}, \tilde{y}, h)$ be the 3D point on the terrain above that location.

Using that H belongs to the true terrain plane one obtains:

$$N^T (G_E - H) = N^T (p_1 + \lambda R_1 q_1 - H) = 0 \quad (60)$$

Extracting λ from (60) and assigning it back to G_E 's expression yields:

$$G_E = p_1 + R_1 \mathcal{L} (H - p_1) \quad (61)$$

For G_E 's uncertainty calculation the derivative of G_E with respect to h should be found:

$$\frac{dG_E}{dh} = R_1 \mathcal{L} \cdot \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T = \frac{R_1 q_1}{N^T R_1 q_1} \quad (62)$$

The above result was obtained using the fact that the z -component of N is 1: $N = \begin{pmatrix} -\nabla DTM & 1 \end{pmatrix}^T$. Finally, the uncertainty of G_E is expressed by the following covariance-matrix:

$$\Sigma_{G_i} = \left(\frac{dG_E}{dh} \right) \cdot \sigma_h^2 \cdot \left(\frac{dG_E}{dh} \right)^T = \sigma_h^2 \cdot \frac{R_1 q_1 q_1^T R_1^T}{(N^T R_1 q_1)^2} \quad (63)$$

5.4. Σ_{C_2} Calculation

The algorithm presented in this work estimates the pose of the first camera frame and the ego-motion. Usually, the most interesting parameters for navigation purpose will be the second camera frame since it reflect the most updated information about the platform location. The second pose can be obtained in a straightforward manner as the composition of the first frame pose together with the camera ego-motion:

$$p_2 = p_1 - R_1 R_{12}^T p_{12} \quad (64)$$

$$R_2 = R_1 R_{12}^T \quad (65)$$

The uncertainty of the second pose estimates will be described by a 6×6 covariance matrix that can be derived from the already obtained 12×12 covariance matrix Σ_θ by multiplication from both sides with J_{C_2} . The last notation is the Jacobian of the six C_2 parameters with respect to the twelve parameters mentioned above. For this purpose, the three Euler angles ϕ_2 , θ_2 and ψ_2 need to be extracted from (65) using the following equations:

$$\phi_2 = \arctan \left(\frac{R_2(2, 3)}{R_2(3, 3)} \right) \quad (66)$$

$$\theta_2 = \arcsin(-R_2(1, 3)) \quad (67)$$

$$\psi_2 = \arctan \left(\frac{R_2(1, 2)}{R_2(1, 1)} \right) \quad (68)$$

Simple derivations and then concatenation of the above expressions yields the required Jacobian which is used to propagate the uncertainty from C_1 and the ego-motion to C_2 . The found covariance matrix Σ_{C_2} is the same as measurement covariant matrix R_k described in section about Kalman filter.

$$R_k = \Sigma_{C_2} \quad (69)$$

6. Divergence of the method. Necessary thresholds for the method convergence.

In previous Section we considered Error analysis for video navigation method. But its consideration is correct only if found solution is close to true one. If it is not true nonlinear effects can appear or even we can found incorrect local minimum. In this case the method can begins to diverge. We can obtain the such result:

1)if large number of outliers features appears.

2)if the case is close to degenerated one. In this case the position or orientation errors are too large. It can happen for example for small number of features, flat ground, small field of view of camera and all that.

3) if the initial position and orientation for iterations process are too far from true values

In the follow subsections we consider some threshold conditions which allow us to avoid the such situations.

If in some case even one of these threshold conditions is not correct we don't use for this case the correction of visual navigation method and use only usual INS result.If such situation repeats three times we stope to use the visual navigation method at all and don't use it also for the last correct case. Let us discourse these three factors in details

6.1 Dealing with Outliers

In order to handle real data, a procedure for dealing with outliers must be included in the implementation. The objective of the present section is to describe the current implementation, which seems to work satisfactorily in practice. Three kinds of outliers should be considered:

1. Outliers present in the correspondence solution (i.e., "wrong matches").
2. Outliers caused by the terrain shape, and
3. Outliers caused by relatively large errors between the DTM and the observed terrain.

The latter two kinds of outliers are illustrated in Fig.7. The outliers caused by the terrain shape appear for terrain features located close to large depth variations. For example, consider two hills, one closer to the camera, the other farther away, and a terrain feature Q located on the closer hill. The ray-tracing algorithm using the erroneous pose may "miss" the proximal hill and erroneously place the feature on the distal one. Needless to say, the error between the true and estimated locations is not covered by the linearization. To visualize the errors introduced by a relatively large DTM-actual terrain mismatch, suppose a building was present on the terrain when the DTM was acquired, but is no longer there when the experiment takes place. The ray-tracing algorithm will locate the feature on the building although the true terrain-feature belongs to a background that is now visible.

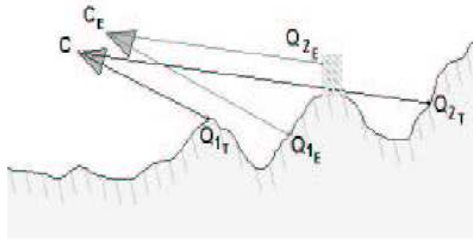


Figure 7: Outliers caused by terrain shape and DTM mismatch. C_T and C_E are true and estimated camera frames, respectively. Q_{1_E} and Q_{2_E} are outliers caused by terrain shape and by terrain/DTM mismatch, respectively.

As discussed above, the multi-feature constraint is solved in a least-squares sense for the pose and motion variables. Given the sensitivity of least-squares to incorrect data, the inclusion of one or more outliers may result in the convergence to a wrong solution. A possible way to circumvent this difficulty is by using an M-estimator, in which the original solution is replaced by a weighted version. In this version, a small weight is assigned to the constraints involving outliers, thereby minimizing their effect on the solution. More specifically, consider the function $f_i(\Theta)$ defined in (45) resulting from the i -th correspondence pair. In the absence of noise, this function should be equal to zero at the true pose and motion values and hence, following standard notation, define the residual $r_i(\Theta) \doteq \|f_i(\Theta)\|$. Using an M-estimator, the solution for Θ (the twelve parameters to be estimated) is obtained using an iterative re-weighted least-squares scheme:

$$\Theta = \arg \min \sum_{i=1}^n w_i r_i^2. \quad (70)$$

The weights w_i are recomputed after each iteration according to their corresponding updated residual. In our implementation we used the so-called *Geman-McClure* function, for which the weights are given by:

$$w(x) = \frac{1}{(1+x^2)^2}. \quad (71)$$

The calculated weights are then used to construct a weighted pseudo-inverse matrix that replaces the regular pseudo-inverse J_T appearing in (50). See [18] for further details about M-estimation techniques. Let us define weights matrix W which allows us to decrease influence of outliers

$$\begin{aligned} r_i &= \|f_i(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12})\| \\ medR &= median(x_i) \\ R_i &= w(r_i/medR) \end{aligned} \quad (72)$$

where $i = 1, \dots, n$ and n is number of features.

The weights matrix W ($3n \times 3n$) can be found as follow: for diagonal elements of W we can write : $W_{ii} = R_k$ where k is integer part of $[(i-1)/3 + 1]$. Non-diagonal elements of $W_{ij} = 0$ for $i \neq j$.

Instead equation (50) we use new one:

$$\begin{aligned} JT &= (J_\theta^T W J_\theta)^{-1} J_\theta^T W \\ \Sigma_\theta &= JT \cdot (J_D \Sigma_D J_D^T) \cdot JT^T \end{aligned} \quad (73)$$

If we know two positions of camera and features position in the first photo so we can find the features position on the second photo. If the distance between true position of some correspondent feature on second photo and the position found by previously described method larger than $3\sigma_I$ we would consider the such feature as outlier. Let us define N_i as number of outliers in initial approximation of cameras position and orientation (i.e. before using visual navigation method) and N_f as number of outliers after visual navigation method corrections. The follow conditions let us to avoid too large number of outliers case:

$$\begin{aligned} N_i &\geq N_f \\ \frac{N_f}{N} &< threshold\% \end{aligned} \quad (74)$$

where N is full number of features and $threshold\%$ is some threshold value. We choose it to be equal 0.1 .

6.2 Degenerated case large errors.

For degenerate case the matrix $J_\theta^T W J_\theta$ in equation (73) can be singular. It gives us follow threshold condition:

$$rcond(J_\theta^T W J_\theta) > threshold_{rcond} \quad (75)$$

where $rcond()$ -Matlab function for matrix reciprocal condition number estimate. It is measure for matrix singularity ($0 < rcond() < 1$). Threshold value $threshold_{rcond}$ is chosen to be 10^{-16} .

Degenerated case because of small number of features, flat ground or small field of view of camera gives the follow threshold conditions:

$$\frac{\sqrt{[\Sigma_{C_2}]_{ii}}}{(3\sigma_I/f)h} < threshold_{dist} \quad (76)$$

where $i = x, y, z$ coordinate indexes for diagonal elements of covariance matrix Σ_{C_2} . $f = 1$ is a focus length of the camera, h is height of the camera. $\frac{3\sigma_I}{f}h$ gives us the maximum camera position shift allowing the photo feature error to be smaller than pixel size. Threshold value $threshold_{dist}$ is chosen to be 40.

$$3\sqrt{[\Sigma_{C_2}]_{ii}} < L_{ground-dist} \quad (77)$$

where $i = x, y, z$ coordinate indexes for diagonal elements of covariance matrix Σ_{C_2} , $L_{ground-dist}$ is character size of ground relief change.

$$\frac{\sqrt{[\Sigma_{C_2}]_{ii}}}{(3\sigma_I/f)} < threshold_{angle} \quad (78)$$

where $i = \phi, \theta, \psi$ angular indexes for diagonal elements of covariance matrix Σ_{C_2} . $\frac{3\sigma_I}{f}$ gives us the maximum camera angular shift allowing the photo feature error to be smaller than pixel size. Threshold value $threshold_{angle}$ is chosen to be 40.

$$3\sqrt{[\Sigma_{C_2}]_{ii}} < \frac{L_{ground-dist}}{h} \quad (79)$$

where $i = \phi, \theta, \psi$ angular indexes for diagonal elements of covariance matrix Σ_{C_2} .

Degenerated case because of small baseline (distance between two camera positions used in video navigation method) gives the follow threshold conditions:

$$\frac{\sqrt{[\Sigma_\theta]_{ii}}}{\|p_{12}\|} < threshold_{dist12} \quad (80)$$

where $i = x_{12}, y_{12}, z_{12}$ mutual coordinate indexes for diagonal elements of covariance matrix Σ_θ . Threshold value $threshold_{dist12}$ is chosen to be 0.1 .

$$\frac{\sqrt{[\Sigma_\theta]_{ii}}}{(\|p_{12}\|/h)} < threshold_{angle12} \quad (81)$$

where $i = \phi_{12}, \theta_{12}, \psi_{12}$ mutual angular indexes for diagonal elements of covariance matrix Σ_θ . Threshold value $threshold_{angle12}$ is chosen to be 0.1 .

6.3 The initial state of the camera is too far from the its true or final calculated state.

Let us define threshold conditions to avoid the initial state of the camera to be too far from the its true state. P_k^- is covariant matrix obtained from INS and previous corrections of INS by video navigation method with help of Kalman filter and described in section about Kalman filter.

$$3\sqrt{[P_k^-]_{ii}} < L_{ground-dist} \quad (82)$$

where $i = x, y, z$ coordinate indexes for diagonal elements of covariance matrix P_k^- .

$$3\sqrt{[P_k^-]_{ii}} < \frac{L_{ground-dist}}{h} \quad (83)$$

where $i = \phi, \theta, \psi$ angular indexes for diagonal elements of covariance matrix P_k^- .

Let us define threshold conditions to avoid the initial state of the camera to be too far from the its final state. The follow four equations give us differences between initial and final state obtain as corrections of INS by video navigation method with help of Kalman filter.

$$\delta p_2 = |p_{2final} - p_{2init}| \quad (84)$$

$$\delta p_{12} = |p_{12final} - p_{12init}| \quad (85)$$

$$\delta \alpha_2 = |\alpha_{2final} - \alpha_{2init}| \mod (2\pi) \quad (86)$$

$$\delta \alpha_{12} = |\alpha_{12final} - \alpha_{12init}| \mod (2\pi) \quad (87)$$

$$3(\sqrt{[P_k^-]_{ii}} + \sqrt{[\Sigma_{C_2}]_{ii}}) > \delta p_{2i} \quad (88)$$

where $i = x, y, z$ coordinate indexes for diagonal elements of covariance matrix P_k^- and Σ_{C_2}

$$3(\sqrt{[P_k^-]_{ii}} + \sqrt{[\Sigma_{C_2}]_{ii}}) > \delta \alpha_{2i} \quad (89)$$

where $i = \phi, \theta, \psi$ angular indexes for diagonal elements of covariance matrix P_k^- and Σ_{C_2}

$$\frac{\delta p_{12i}}{\|p_{12}\|} < threshold_{dist12} \quad (90)$$

where $i = x_{12}, y_{12}, z_{12}$ mutual coordinate indexes.

$$\frac{\delta \alpha_{12i}}{(\|p_{12}\|/h)} < threshold_{angle12} \quad (91)$$

where $i = \phi_{12}, \theta_{12}, \psi_{12}$ mutual angular indexes.

7. Simulations Results

7.1 Dependence of error analysis on different factors.

The purpose of the following section is to study the influence of different factors on the accuracy of the proposed algorithm estimates. The closed form expression that was developed throughout the previous section is being used to determine the uncertainty of these estimates under a variety of simulated scenarios. Each tested scenario is characterized by the following parameters: the number of optical-flow features being used by the algorithm, the image resolution, the grid spacing of the DTM (also referred as the DTM resolution), the amplitude of hills/mountains on the observed terrain, and the magnitude of the ego-motion components. At each simulation, all parameters except the examined one are set according to a predefined parameters set. In this *default scenario*, a camera with 400×400 image resolution flies at altitude of 500m above the terrain. The terrain model dimensions are 3×3 km with 300m elevation differences (Fig.13(b)). A DTM of 30m grid spacing is being used to model the terrain (Fig.10(c)). The DTM resolution leads to a standard-deviation of 2.34m for the height measurements. The default-scenario also defines the number of optical-flow features to about 170, where an ego-motion of $\|p_{12}\| = 40m$ and $\|(\phi_{12}, \theta_{12}, \psi_{12})\| = 10^\circ$ differs the two images being used for the optical-flow computation. Each of the simulations described below study the influence of different parameter. A variety of values are examined and 150 random tests are performed for each tested value. For each test the camera position and orientation were randomly selected, except the camera's height that was dictated by the scenario's parameters. Additionally, the direction of the ego-motion translation and rotation components were first chosen at random and then normalized to the require magnitude.

In Fig.8, the first simulation results are presented. In this simulation the number of optical-flow features that are used by the algorithm is varied and its influence on the obtained accuracy of C_2 and the ego-motion is studied. All parameters were set to their default values except for the features number. Fig.8(a) presents the standard-deviations of the second frame of the camera while the deviations of the ego-motion are shown in Fig.8(b). As expected, the accuracy improves as the number of features increases, although the improvement becomes negligible after the features' number reaches about 150.

In the second simulation the influence of the image resolution was studied (Fig.9). It was assumed that the image measurements contain uncertainty of half-pixel, where the size of the pixels is dictated by the image resolution. Obviously, the accuracy improves as image resolution increases since the quality of the optical-flow data is directly depends on this parameter.

The influence of DTM grid spacing is the objective of the next simulation. Different DTM resolutions were tested varying from 10m up to an extremely rough resolution of 190m between adjacent grid points (see Fig.10). The readers attention is drawn to the fact that the obtained accuracy seems to decrease linearly with respect to the DTM grid-spacing (see Fig.11). This phenomenon can be understood since, as was explained in the previous section, the DTM resolution does not affect the accuracy directly but rather it influences the height uncertainty which is involved in the accuracy calculation. As can be seen in Fig.12, the standard-deviation of the DTM heights increases linearly with respect to the DTM grid spacing which is the reason for the obtained results.

Another simulation demonstrates the importance of the terrain structure to the estimates accuracy. In the extreme scenario of flying above a planar terrain, the observed ground features do not contain the required information for the camera pose

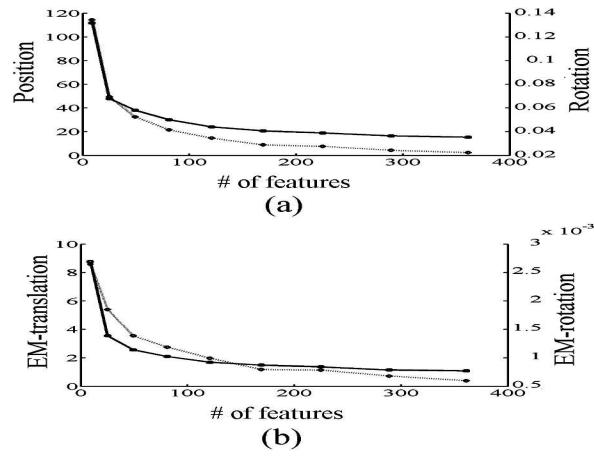


Figure 8: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the number of flow-features. In both graphs, the left vertical axis measures the translational deviations (in meters) and corresponds to the solid graph-line, while the right vertical axis measures the rotational deviations (in radians) and corresponds to the dotted graph-line

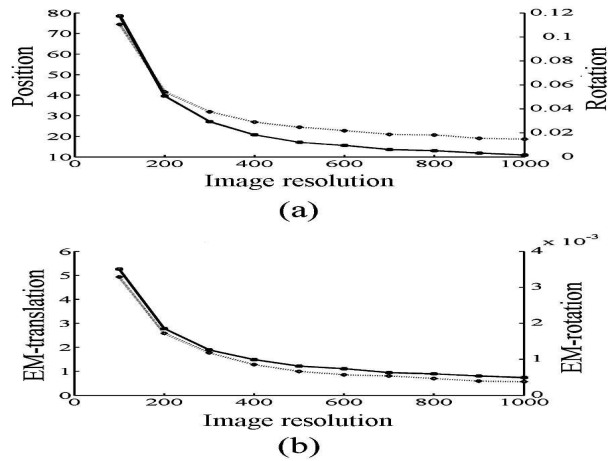


Figure 9: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the image resolution

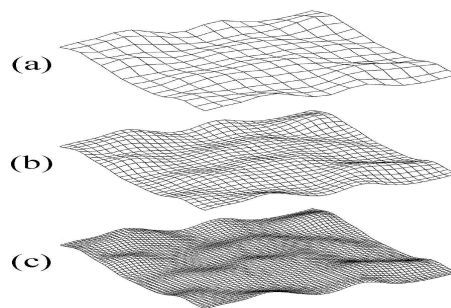


Figure 10: Different DTM resolutions: (a) grid spacing = 190m, (b) grid spacing = 100m, (c) grid spacing = 30m

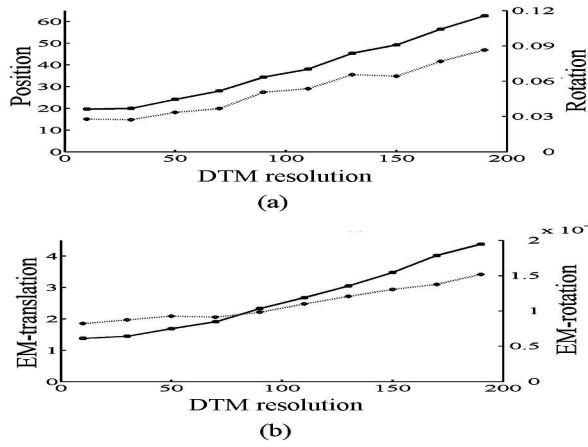


Figure 11: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the grid-spacing of the DTM

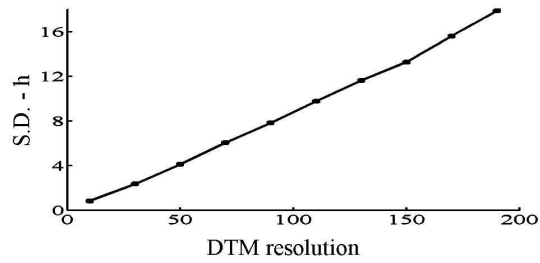


Figure 12: standard-deviation of the DTM's height measurement with respect to the grid-spacing of the DTM

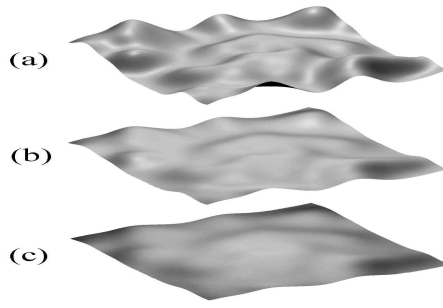


Figure 13: DTM elevation differences: (a) 150m, (b) 300m, (c) 450m

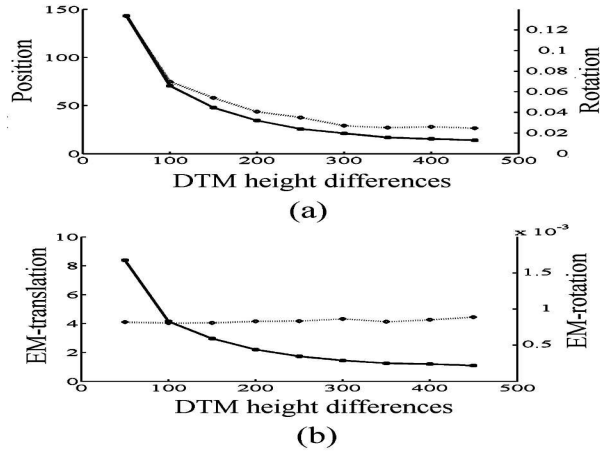


Figure 14: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the height differences of the terrain

derivation, and a singular system will be obtained. As the height differences and the variability of the terrain increase, the features become more informative and a better estimates can be derived. For this simulation, the DTM elevation differences were scaled to vary from 50m to 450m (Fig.13). It is emphasized that while the terrain structure plays a crucial role at the camera pose estimation together with the translational component of the ego-motion, it has no direct affect on the ego-motion rotational component. As the optical-flow is a composition of two vector fields - translation and rotation, the information for deriving the ego-motion rotation is embedded only in the rotational component of the flow-field. Since the features depths influence only the flow's translational component it is expected that the varying height differences or any other structural change in the terrain will have no affect on the ego-motion rotation estimation. The above characteristics are well demonstrated in Fig.14.

Since it is the translation component of the flow which holds the information required for the pose determination, it would be interesting to observe the effect of increasing the magnitude of this component. The last simulation presented in this work demonstrates the obtained pose accuracy when the ego-motion translation component vary form 5m to 95m. Although it has no significant effect on the ego-motion accuracy, the uncertainty of the pose estimates decreases for a large magnitude of translations (see Fig.15). As a conclusion from the above stated, the time gap between the two camera frames should be as long as the optical-flow derivation algorithm can tolerate.

7.2 Results of numerical simulation for real parameters of flight and camera.

Inertial navigation systems (INS) are used usually for detection of missile position and orientation. The problem of this method is that its error increases all time. We propose to use new method (Navigation Algorithm based on Optical-Flow and a Digital Terrain Map) [15] to correct result of INS and to make the error to be finite and constant. Kalman Filter is used to combine results of INS and results of new method [12]. Error analysis with linear first-order approximation is used to find

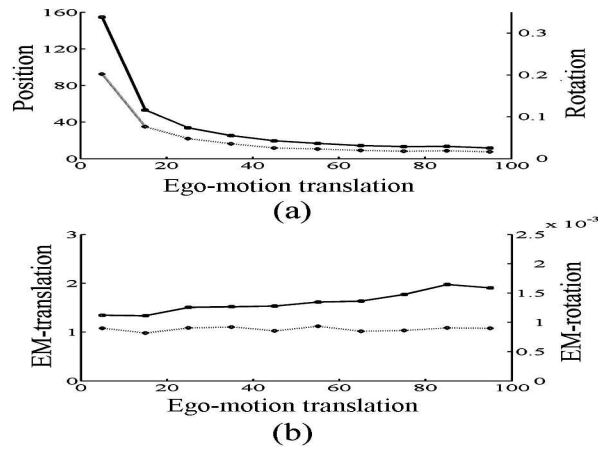


Figure 15: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the magnitude of the translational component of the ego-motion

error correlation matrix for our new method [14]. We made numerical simulations of flight with real parameters of flight and camera using only INS and INS and our new method to check usefulness of this new method.

The chosen flight parameters are following:

- Height of flight is 700, 1000, 3000 m.
- Velocity of flight is 200m/s.
- Flight time is 800 s.

Trajectory of the flight we can see on (Fig.16). Digital Terrain Map of real ground was used as cell (Fig.17) for our simulations. This cell was continued periodically to obtain full Map of the ground (Fig.18). Random noise was used as main component of INS noise. The more real drift and bias noise give much bigger mistake (about 6000 m instead 1000 m in the finish point of the flight).

The chosen camera and simulation parameters are following:

- FOV (field of view of camera) is 60 degree. (FOV is field of view of camera.)
- Features number found on photos is 100, 120.
- Resolution of camera is 500x500, 1000x1000, 4000x4000.(The resolution of camera defines precision of feature detection, we assume no Optical Flow outliers for features.)
- Baseline is 30m, 50m or 200m. (Baseline is distance between two camera positions used to make two photos for new method.)
- $\Delta time$ is 5s, 15 s, 30s.($\Delta time$ is time interval between measurements.)

The typical results of numerical simulations can be seen on (Fig.3, 4, 5, 6) for different cases of flight, camera and simulation parameters. Let us demonstrate error tables for typical case with positive results: x, y, z position errors of INS with using new method and without using new method.

Used flight, camera and simulation parameters for this case:

- FOV is 60 degree
- Number of features is 120

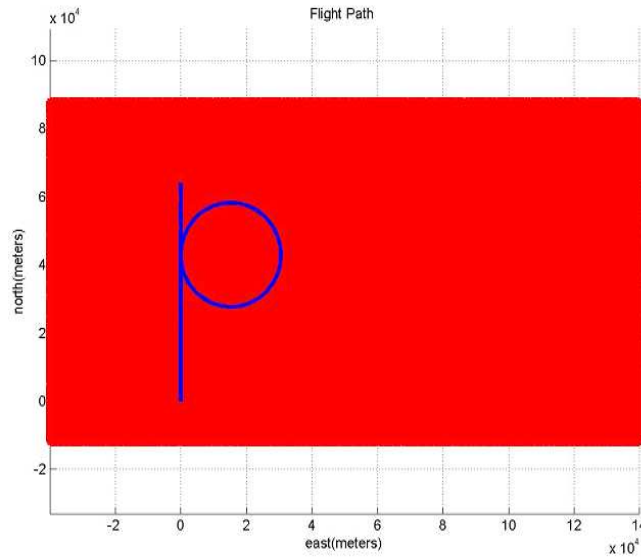


Figure 16: Trajectory of the flight.

Resolution is 1000x1000
 Baseline is 200m
 $\Delta time$ is 15 s.
 Flight velocity is 200 m/s
 Heights are 700m, 1000 m, 3000m.

Height	700m	1000m	3000m
Max x error without new method	900m	130m	1300 m
Max x error with new method	25 m	20 m	100 m

Table 1. x axis max error for INS with and without new method for different heights.

Height	700m	1000m	3000m
Max y error without new method	1000m	2000m	400m
Max y error with new method	25m	20m	100 m

Table 2. y axis max error for INS with and without new method for different heights.

Height	700m	1000m	3000m
Max z error without new method	250m	180m	250 m
Max z error with new method	25m	20m	150m

Table 3. z axis max error for INS with and without new method for different heights.

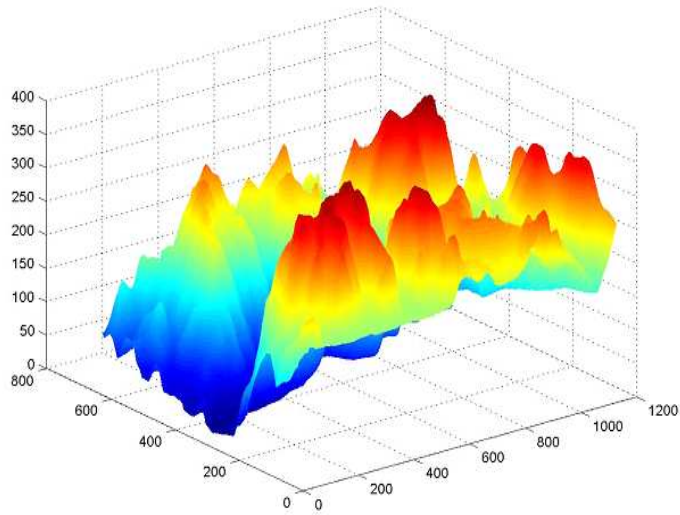


Figure 17: Map of real ground was used as cell.

Let us demonstrate error tables for typical case with positive results: x, y, z position errors of INS with using new method for different resolutions of camera. Used flight, camera and simulation parameters for this case:

FOV 60 degree, Number of features:120, Resolution 500x500, 1000x1000, 4000x4000, Baseline 200m, Deltatime 15 s, Flight velocity 200 m/s, Heights: 1000 m.

Resolution	500x 500	1000x 1000	4000x 4000
Max x error with new method	50m	20m	10m

Table 4. x axis max error for INS with new method for different resolutions of camera.

Resolution	500x 500	1000x 1000	4000x 4000
Max y error with new method	50m	20m	10m

Table 5. y axis max error for INS with new method for different resolutions of camera.

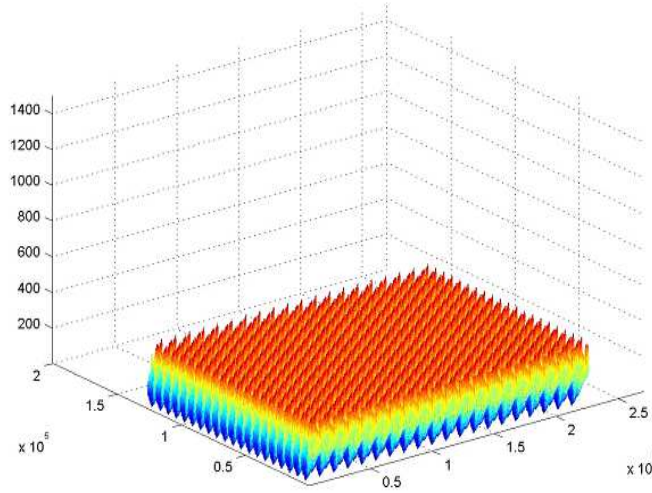


Figure 18: cell was continued periodically to obtain full Map of the ground.

Resolution	500x	1000x	4000x
	500	1000	4000
Max z error with new method	35m	20m	10m

Table 6. z axis max error for INS with new method for different resolutions of camera.

8 Open problems and future method development.

1) If situation is close to degenerated case (for example, for small camera field of view, almost flat ground, small baseline and so on) we can not use described method because it is impossible to find camera states from this data. But it is possible also for this case to use found correspondent features constrains for INS results improvement by help Kalman filter. We can consider directly these correspondent features (and not calculated position and orientation on basis these features) as result of measurement for Kalman filter. Example of the such improvement can be found in [16]. But in this case errors of method will increase with time similar to INS. So after some time measured position is too far from the true position and we can not use DTM constrains for error correction, but only epipolar constrains. For described in this paper method the error stops to increase and remains constant so we are capable to use DTM constrains all time.

2) It is possible to consider more optimal and fast methods for looking for minimum of function giving position and orientation of camera. For example it is possible to improve initial state for described method, using epipolar equations (25) for R_{12} and p_{12} up to constant calculations. The next step can be use equation (21) for R_1 calculation. And final step using equation (18) for p_{12} and p_1 calculation. The result can be improved by described iteration method.

3) We can look for not only some random features. Also hill tops, valleys and hill occluding boundaries can be used for position and orientation specifying.

4) using distributed (not point) features and also some character object recognition.

5) Using the used methods in different practical situations: orientation in rooms, inside of man body.

9 Conclusions

An algorithm for pose and motion estimation using corresponding features in images and a DTM was presented with using Kalman filter. The DTM served as a global reference and its data was used for recovering the absolute position and orientation of the camera. In numerical simulations position and velocity estimates were found to be sufficiently accurate in order to bound the accumulated errors and to prevent trajectory drifts.

An error analysis has been performed for a novel algorithm that uses as input the optical flow derived from two consecutive frames and a DTM. The position, orientation and ego-motion parameters of the camera can be estimated by the proposed algorithm. The main source for errors were identified to be the optical-flow computation, the quality of the information about the terrain, the structure of the observed terrain and the trajectory of the camera. A closed form expression for the uncertainty of the pose and motion was developed. Extensive numerical simulations were performed to study the influence of the above factors.

Tested under reasonable and common scenarios, the algorithm behaved robustly even when confronted with relatively noisy and challenging environment. Following the analysis, it is concluded that the proposed algorithm can be effectively used as part of a navigation system of autonomous vehicles.

On basis results of numerical simulation for real parameters of flight and camera we also can conclude follow:

1) The most important parameter of simulations is FOV: for the small FOV the method diverges. For FOV 60 degree the results are very good. The reason for this is that for small FOV (12 or 6 degree) the situation is close to degenerated state, also we must choose small baseline and observed ground patch is too small and almost flat.

2) Resolution of camera is also very important parameter: for better resolution we have much more better results, because of much more better precision of features detection.

3) The precision of new method depends on flight height. Initially precision increases with height increasing because we can use bigger baseline and can see bigger patch of ground. But for bigger heights precision begin to decrease because of small parallax effect.

Acknowledgment

We would like to thank Ronen Lerner, Ehud Rivlin and Hector Rotstein for very useful consultations.

References

- [1] John Oliensis, A critique of structure-from-motion algorithms, *Computer Vision and Image Understanding*, vol. 80, pp. 1722-14, 2000
- [2] Y. Liu, M.A. Rodrigues, "Statistical image analysis for pose estimation without point correspondences", *Pattern Recognition Letters*, vol. 22, pp. 1191-1206, 2001
- [3] P. David, D. DeMenthon, R. Duraiswami, H. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination", *ECCV 2002, LNCS 2352*, pp. 698-714, 2002
- [4] J. L. Barron and R. Eagleon, "Recursive estimation of time-varying motion and structure Parameters", *Pattern Recognition* vol. 29, no. 5, pp. 797-818, 1996
- [5] T.Y. Tian, C. Tomashi, D.J. Hegger, "Comparison of approaches to egomotion computation", *Department of Psychology and Computer science, Stanford university, CA 94305*, 1996
- [6] A.Chiuso, S.Soatto, "MFm: 3-D Motion From 2-D Motion, Causally integrated over time", *Washington University Technical Report*, 1999
- [7] M. Irani, B. Rousso, S. Peleg, "Robust Recovery of Ego-Motion", *Proc. Of CAIP 93*, pp. 371-378, 1993
- [8] D.G. Sim, R.H. Park, R.C. Kim, S.U. Lee, I.C. Kim, "Integrated position estimation using aerial image sequences", *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no.1, 2002
- [9] Technical report (to be added after the double-blind review).
- [10] R. M. Haralick, "Propagating Covariance In Computer Vision", *Advances in Image Understanding*, Azriel Rosenfeld, Bowyer and Ahuja, (eds.), *IEEE Computer Society Press*, pp. 142-157, Washington, 1996

- [11] Rees W.G., "The accuracy of Digital Elevation Models interpolated to higher resolutions", *International Journal of Remote Sensing* vol. 21 no.1, pp. 7-20, 2000
- [12] Greg Welch and Gary Bishop "An Introduction to the Kalman Filter" UNC-Chapel Hill TR 95-041,NC 27599-3175 April 5, 2004
www.menem.com/ilya/digital_library/control/welch-bishop-01.pdf
- [13] www.GPSSoftNav.com
- [14] O. Kupervasser, R. Lerner, E.Rivlin and P. H. Rotstein "Error Analysis for a Navigation Algorithm based on Optical-Flow and a Digital Terrain Map" *Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium*, P.1203-1212
- [15] R. Lerner, E.Rivlin and P. H. Rotstein "Pose and Motion Recovery from Feature Correspondences and a Digital Terrain Map" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1404-1417, Sept., 2006.
- [16] Pini Gurfil,Hector Rotstein "Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach", *Journal of Guidance, Control and Dynamics* 0731-5090 vol.24 no.5 (1016-1028),2001
- [17] C. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 22(6):610622, 2000.
- [18] D. Hoaglin, F. Mosteller, and J. Tukey, editors. *Understanding robust and exploratory data analysis*. John Wiley & Sons Inc, 1983.

КОРРЕКЦИЯ ОШИБОК ИНЕРЦИАЛЬНОЙ НАВИГАЦИОННОЙ СИСТЕМЫ С ПОМОЩЬЮ МЕТОДОВ ВИЗУАЛЬНОЙ НАВИГАЦИИ

В.В.Воронов

ЗАО «Р.Е.Т.Кронштадт», Россия

107076, Россия, Москва,

Ул. Стромьнка, д.18

Тел. +7(495)748 3584

voronov@kronshtadt.ru

О.Ю.Купервассер

ЗАО «Р.Е.Т.Кронштадт», Россия

107076, Россия, Москва,

Ул. Стромьнка, д.18

Тел. +7(499)1343965

olegkup@yahoo.com

Предложен алгоритм для нахождения позиции, ориентации и оценки движения, использующий соответствующие точки в изображениях и цифровую карту ландшафта. Использование Цифровой Карты Ландшафта (ЦКЛ) как глобальной справочной информации позволяет восстановление абсолютной позиции и ориентации камеры. Чтобы сделать это, ЦКЛ используется, чтобы сформулировать дополнительные ограничения между соответствующими точками в двух последовательных кадрах. Использование этих данных позволяет улучшить надежность и точность инерциального навигационного алгоритма. Расширенный фильтр Калмана использовался, чтобы объединить результаты инерциального навигационного алгоритма и навигационного алгоритма, основанного на компьютерном зрении. Выполнимость этого алгоритма продемонстрирована путем численного моделирования.

1. Введение

Алгоритмы визуальной навигации являются одним из основных объектов исследований в течении последнего десятилетия. Бурное развитие роботизированных систем потребовало выработки специфических подходов для определения в реальном времени координат местоположения и параметров ориентации подвижных объектов, как в закрытых помещениях, так и на улице. Одним из важных требований к современным беспилотным авиационным комплексам является возможность осуществления точной навигации в условиях отсутствия спутниковой информации. Интеграция визуальных методов навигации с традиционными инерциальными системами может обеспечить устойчивое решение навигационной задачи для наземных и летающих автоматических систем.

Существуют три основных подхода к решению задачи определения параметров движения объекта по визуальной информации.

1) Определение параметров движения и ориентации путем отслеживания перемещения в кадре изображений неких характерных (особых) точек местности [2,3]. После идентификации особых точек на каждом снимке точное определение параметров движения представляет собой достаточно тривиальную задачу. Наиболее сложной проблемой остаётся детектирование особых точек и их соотнесение с набором характерных признаков данной местности, определенным заранее.

2) При подходе “ego motion” [4-7] оценивается относительное движение камеры по отношению к предыдущему ее положению. Источником данных являются видеокдры. Оценив параметры движения и зная начальные условия, можно путём интегрирования вычислить траекторию движения камеры. Важным достоинством этого метода (в отличие от предыдущего) является отсутствие необходимости выявления особых точек на местности. Проблемной стороной является нарастание ошибки в

процессе интегрирования. Таким образом, имеется аналог «дрейфа» оценки местоположения, и точность оценивания деградирует с течением времени (или с ростом пройденного пути).

Два изложенных подхода объединяет необходимость анализа последовательных изображений.

3) Третий метод позволяет компенсировать упомянутый выше эффект «дрейфа», применяя дополнительный алгоритм прямого оценивания местоположения. В представляемой работе используются данные цифрового рельефа местности (ЦРМ). ЦРМ представляет собой цифровое представление топографии местности через матрицу высот над уровнем моря для дискретных координат местоположения. Поставленная задача известна под названием извлечение структуры из движения (structure from motion - SfM) [1]. Восстановленный методами SfM рельеф может быть соотнесен с данными ЦРМ и, таким образом, определяется местоположение объекта относительно ЦРМ. В представляемой работе предлагается непосредственно комплексировать данные ЦРМ и видовые данные [8], полученные с камеры объекта для оценки положения и ориентации объекта, минуя стадию восстановления рельефа по видеоданным. В результате сокращается объём вычислений, и повышается устойчивость алгоритма.

Таким образом, имеются несколько решений навигационной задачи через методы видовой обработки данных. Каждый из предложенных методов базируется на уникальной априорной информации и даёт независимое решение с той или иной степенью точности. Любое из этих решений, как и все вместе, может использоваться для коррекции инерциальной навигационной системы с применением стандартной техники интегрирования данных посредством Калмановской фильтрации.

В представляемой работе изложены алгоритмы получения навигационных данных из видовой информации с использованием ЦРМ, рассмотрены вопросы комплексирования видовых данных с данными инерциальной системы, приводятся результаты компьютерного моделирования.

2. Определение и описание проблемы.

Проблема может быть кратко описана следующим образом: В любой момент времени t , система координат $C(t)$ установлена на камеру таким способом, что Z -ось совпадает с оптической осью, и начало координат совпадает с центром проектирования камеры. В этот момент времени камера расположена в некотором географическом местоположении $p(t)$ и имеет данную ориентацию $R(t)$ относительно глобальной координатной системы W ($p(t)$ - трехмерный вектор, $R(t)$ - ортонормальная матрица вращения). $p(t)$ и $R(t)$ определяют преобразование из координатной системы камеры $C(t)$ в глобальную координатную систему W , где, если ${}^c v$ и ${}^w v$ являются векторами в $C(t)$ и W соответственно, то ${}^w v = R(t) {}^c v + p(t)$.

Рассмотрим теперь два последовательных момента времени t_1 и t_2 : преобразование из $C(t_1)$ в $C(t_2)$ дано вектором сдвига $\Delta p(t_1, t_2)$ и матрицей вращения $\Delta R(t_1, t_2)$ таким образом, что ${}^{c(t_2)} v = \Delta R(t_1, t_2) {}^{c(t_1)} v + \Delta p(t_1, t_2)$. Для грубой оценки позы камеры в t_1 и собственное движение камеры между двумя моментами времени - $p_E(t_1)$, $R_E(t_1)$, $\Delta p_E(t_1, t_2)$ и $\Delta R_E(t_1, t_2)$ используются. (Символ "E" для нижнего индекса обозначает, что это оценочная (estimated) величина.)

Также используется поле оптического потока: $\{u_i(t_k)\}$ ($i=1\dots n, k=1,2$). Для i 'th характерная точка местности, $u_i(t_1) \in \mathbb{R}^2$ и $u_i(t_2) \in \mathbb{R}^2$ представляют её местоположение на первом и втором кадре соответственно.

Используя вышеупомянутые обозначения, цель предложенного алгоритма состоит в том, чтобы оценить истинной позу камеры и её собственное движение: $p(t_1)$, $R(t_1)$,

$\Delta p(t_1, t_2)$ и $\Delta R(t_1, t_2)$ используя поле оптического потока $\{u_i(t_k)\}$, ЦКЛ и приблизительные начальные условия: $p_E(t_1)$, $R_E(t_1)$, $\Delta p_E(t_1, t_2)$ и $\Delta R_E(t_1, t_2)$.

3. Навигационный алгоритм

Следующий раздел описывает навигационный алгоритм, который оценивает вышеупомянутые параметры. Поза и движение эго-камеры получены, используя ЦКЛ и поле оптического потока для двух последовательных кадров. В отличие от метода наземных ориентиров никакие характерные ориентиры не должны быть обнаружены и распознаны. Только соответствие между двумя последовательными изображениями должно быть найдено, чтобы получить поле оптического потока. Как было упомянуто в предыдущем разделе, грубая оценка искомых параметров используется как первое приближение. Однако, так как алгоритм только использует это как начальное приближение и повторно вычисляет позу и движение непосредственно, никакое сложение предыдущих ошибок не будет иметь место, и точность будет сохраняться.

Новый подход основан на следующем наблюдении. Так как ЦКЛ предоставляет информацию о структуре наблюдаемого ландшафта, глубину наблюдаемых характерных точек местности определяет поза камеры. Следовательно, учитывая позу и движение камеры, поле оптического потока может быть однозначно определено. Целью алгоритма является нахождение позы и движения, которые приводят к полю оптического потока наиболее близкому, насколько это возможно, к найденному выше полю оптического потока.

Единичный вектор из поля оптического потока будет использоваться, чтобы определить ограничения на позу камеры и её движение. Пусть ${}^w G \in \mathbb{R}^3$ является местоположением характерной точки местности в трехмерном пространстве. В два различных момента времени t_1 и t_2 , эта характерная точки местности проектируется на плоскость изображения камеры в точки $u(t_1)$ и $u(t_2)$. Используя модель дырочной камеры, получаем $u(t_1), u(t_2) \in \mathbb{R}^2$. Позвольте ${}^c q(t_1)$ and ${}^c q(t_2)$ быть гомогенными представлениями этих местоположений. Также можно описать эти вектора как вектора из оптического центров камер к точкам проектирования на плоскости изображений. Используя начальную оценку позы камеры в t_1 , линия, проходящая через $p_E(t_1)$ и ${}^c q(t_1)$, может быть пересечена с ЦКЛ. Любой алгоритм трассировки лучей может использоваться для этой цели. Местоположение этого пересечения обозначено как ${}^w G_E$. Символ "E" нижнего индекса проясняет тот факт, что эта точка местности - предполагаемое местоположение для характерной точки, который вообще будет отличаться от истинного местоположения характерной точки ${}^w G$. Различие между истинным и оцененным местоположением происходит из двух основных источников: ошибка в начальном предположении для позы и ошибка в определении ${}^w G_E$, вызванной дискретизацией ЦКЛ и основными погрешностями. Для разумных ошибок начального положения и ЦКЛ - связанных ошибок, две точки ${}^w G_E$ и ${}^w G$ должны быть достаточно близким, чтобы позволить линеаризацию ЦКЛ вокруг ${}^w G_E$. Обозначая N нормаль плоскости, касательной к ДТМ в точке ${}^w G_E$, можно написать:

$$N^T ({}^w G - {}^w G_E) \approx 0 \quad (1)$$

Истинная характерная точка местности ${}^w G$ может быть описана, используя истинные параметры позы:

$${}^w G = R(t_1) \cdot {}^c q(t_1) \cdot \lambda + p(t_1) \quad (2)$$

Здесь, λ обозначает глубину характерной точки (то есть расстояние от точки до плоскости изображения, спроектированное на оптическую ось). Подставляя (2) в (1):

$$N^T (\lambda \cdot R(t_1) \cdot {}^c q(t_1) + p(t_1) - {}^w G_E) = 0 \quad (3)$$

Из этого выражения глубина истинной характерной точки может быть рассчитана, используя оценочное положение характерной точки:

$$\lambda = \frac{N^T {}^w G_E - N^T p(t_1)}{N^T R(t_1) \cdot {}^c q(t_1)} \quad (4)$$

Подстановкой (4) обратно в (2) получаем:

$${}^w G = R(t_1) \cdot {}^c q(t_1) \cdot \left(\frac{N^T {}^w G_E - N^T p(t_1)}{N^T R(t_1) \cdot {}^c q(t_1)} \right) + p(t_1) \quad (5)$$

Чтобы упростить обозначения, $R(t_i)$ будет заменен R_i и аналогично для $p(t_i)$ и $q(t_i)$ $i=1,2$. $\Delta R(t_1, t_2)$ и $\Delta p(t_1, t_2)$ будут заменены на R_{12} и p_{12} соответственно. Верхний индекс, описывающий систему координат, в которой дан вектор, будет также опущен, за исключением случаев, где требуется особое внимание к описываемым системам координат. Обычно, p_{12} и q 's находится в системе координат камеры, в то время как остальная часть векторов дана в глобальной системе координат. Используя упрощенные обозначения, (5) может быть переписано как:

$$G = \frac{R_1 q_1 N^T}{N^T R_1 q_1} G_E - \frac{R_1 q_1 N^T}{N^T R_1 q_1} p_1 + p_1 \quad (6)$$

Для того, чтобы получить более простое выражение определим следующий проектирующий оператор:

$$\mathcal{P}(u, s) = \left(I - \frac{us^T}{s^T u} \right) \quad (7)$$

Этот оператор проектирует вектор на нормаль подпространства к s , вдоль направления u . Как иллюстрация, это просто проверить, что $s^T \cdot \mathcal{P}(u, s)v \equiv 0$ и $\mathcal{P}(u, s)u \equiv 0$. Добавляя и вычитая G_E к (6), после переупорядочения:

$$G = G_E + \left[I - \frac{R_1 q_1 N^T}{N^T R_1 q_1} \right] p_1 - \left[I - \frac{R_1 q_1 N^T}{N^T R_1 q_1} \right] G_E \quad (8)$$

Используя проектирующий оператор, (8) становится:

$$G = G_E + \mathcal{P}(R_1 q_1, N)(p_1 - G_E) \quad (9)$$

У вышеописанного выражения есть ясная геометрическая интерпретация (см. Рис.2). Вектор из G_E к p_1 проектируется на касательную плоскость. Проектирование идет вдоль направления $R_1 q_1$, которое является направлением луча из оптического центра камеры (p_1), проходящего через соответствующую точку изображения.

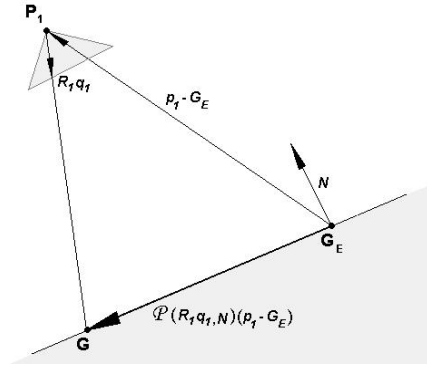


Рис. 1: Геометрическое описание выражения (9), используя проектирующий оператор (7)

Наш следующий шаг будет перенос G из глобальной системы координат - W в систему координат первой камеры C_1 и затем к системе координат второй камеры C_2 . Так как p_1 and R_1 описывают преобразование из C_1 в W , мы будем использовать обратное преобразование:

$${}^{c_2}G = p_{12} + R_{12}(R_1^T(G - p_1)) \quad (10)$$

Подстановка (9) в (10) дает:

$${}^{c_2}G = p_{12} + R_{12}\mathcal{L}(G_E - p_1) \quad (11)$$

\mathcal{L} в написанном выше выражении представляет:

$$\mathcal{L} = \frac{q_1 N^T}{N^T R_1 q_1} \quad (12)$$

Можно интерпретировать \mathcal{L} как оператор обратный к \mathcal{P} : он проектирует векторы на луч, продолжающий $R_1 q_1$ вдоль плоскости, ортогональной к N .

q_2 - проекция истинной характерной точки местности G . Таким образом, векторы q_2 и ${}^{c_2}G$ должны совпасть. Это наблюдение может быть выражено математически, проектируя ${}^{c_2}G$ на продолжение луча q_2 :

$${}^{c_2}G = \frac{q_2}{|q_2|} \cdot \left(\frac{q_2^T}{|q_2|} \cdot {}^{c_2}G \right) \quad (13)$$

В выражении (13), $q_2^T/|q_2| \cdot {}^{c_2}G$ является величиной ${}^{c_2}G$'s прекции на q_2 . Преобразуя (13) и используя проектирующий оператор, мы получаем:

$$\left[I - \frac{q_2 \cdot q_2^T}{q_2^T \cdot q_2} \right] \cdot {}^c G = \mathcal{P}(q_2, q_2) \cdot {}^c G = 0 \quad (14)$$

${}^c G$ является проекцией на ортогональную компонент q_2 . Так как ${}^c G$ и q_2 должны совпадать, это проектирование должно давать нулевой вектор. Подстановка (11) в (14) приводит к нашему окончательному ограничению:

$$\mathcal{P}(q_2, q_2)[p_{12} + R_{12}\mathcal{L}(G_E - p_1)] = 0 \quad (15)$$

Это ограничение включает позицию, ориентацию и собственное движение, определяемое на основе двух кадров камеры. Хотя оно включает трехмерные вектора, ясно, что его ранг не может превысить двойку из-за использования \mathcal{P} , который проектирует \mathbb{R}^3 на двумерное подпространство.

Такое ограничение может быть установлено для каждого вектора в поле оптического потока, пока не будет получена несингулярная система. Так как двенадцать параметров должны быть оценены (шесть для позы и шесть для собственного движения), по крайней мере, шесть векторов оптического потока требуются для решения системы. Но это - правильное заключение для нелинейной проблемы. Если мы используем метод итераций Гаусса-Ньютона, то делаем линеаризацию нашей проблемы около приближенного решения. Найденная матрица будет всегда сингулярной для шести точек (с нулевым детерминантом), как численное моделирование демонстрирует. Таким образом, необходимо использовать, по крайней мере, семь точек, чтобы получить несингулярное линейное приближение. Обычно, чем больше векторов будет использоваться, чтобы определить переопределенную систему, тем более устойчиво решение. Внимание должно быть привлечено к факту, что было получено нелинейное ограничение. Таким образом, итерационная схема будет использоваться, чтобы решить эту систему. Устойчивый алгоритм, который использует итерации Гаусса-Ньютона и М-оценщик описан в [23]. Мы начинаем использовать метод Levenberg-Marquardt, если метод Гаусса-Ньютона после нескольких итераций прекратил сходиться. Эти два алгоритма реализованы в функции lsqnonlin() пакета Matlab. Применимость, точность и надежность алгоритма были проверены через численное моделирование и лабораторные эксперименты.

Более удобно использовать более устойчивое для итераций решение, эквивалентное уравнению (15):

$$\mathcal{P}(q_2, q_2)[p_{12} + R_{12}\mathcal{L}_i(G_{E_i} - p_1)] / |{}^c G| = 0 \quad (16)$$

3.1 Множественные характерные точки.

Предположим, что n характерных точек n отслеживаются при двух положениях камеры так, чтобы их оценочные местоположения Q_{E_i} и их проекции на плоскость изображений q_{1i} и q_{2i} были оценены и измерены, соответственно, для $i = 1, \dots, n$. Нормальный вектор к ДТМ в точке Q_{E_i} обозначим N_i .

Тогда можно переписать (15) в матричной форме как:

$$\begin{bmatrix} -\mathcal{P}(q_{2i}) & \mathcal{P}(q_{2i}) \frac{R_{12} q_{1i} N_i^T}{N_i^T R_1 q_{1i}} \end{bmatrix} \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} =$$

$$\mathcal{P}(q_{2i}) \frac{R_{12} q_{1i} N_i^T}{N_i^T R_1 q_{1i}} Q_{Ei}. \quad (17)$$

Повторяя этого для каждой особенной точки, получаем:

$$\begin{bmatrix} -\mathcal{P}(q_{21}) & \mathcal{P}(q_{21}) \frac{R_{12} q_{11} N_1^T}{N_1^T R_1 q_{11}} \\ -\mathcal{P}(q_{22}) & \mathcal{P}(q_{22}) \frac{R_{12} q_{12} N_2^T}{N_2^T R_1 q_{12}} \\ \vdots & \vdots \\ -\mathcal{P}(q_{2n}) & \mathcal{P}(q_{2n}) \frac{R_{12} q_{1n} N_n^T}{N_n^T R_1 q_{1n}} \end{bmatrix} \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = \begin{bmatrix} \mathcal{P}(q_{21}) \frac{R_{12} q_{11} N_1^T}{N_1^T R_1 q_{11}} Q_{E1} \\ \mathcal{P}(q_{22}) \frac{R_{12} q_{12} N_2^T}{N_2^T R_1 q_{12}} Q_{E2} \\ \vdots \\ \mathcal{P}(q_{2n}) \frac{R_{12} q_{1n} N_n^T}{N_n^T R_1 q_{1n}} Q_{En} \end{bmatrix} \quad (18)$$

В компактной форме:

$$A_n \begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = B_n. \quad (19)$$

Отметим, что A_n и B_n зависят как от известных величин: особенные точки, нормали к касательным плоскостям ЦКЛ, их изображения на двух снимках, так и от неизвестных ориентации R_1 и относительного вращения R_{12} . Теперь сделаем несколько замечаний:

Замечание 1: Связь (18) включает двенадцать "неизвестных", а именно, поза и движение эго-камеры. Из замечания в конце предыдущего раздела следует, что уравнение представляют собой $2n$ линейно-независимых ограничений. Следовательно, чтобы иметь определенную систему уравнений необходимо иметь по крайней мере шесть особенностей в различных местоположениях Q_{Ti} . Но это - правильное заключение для нелинейной задачи. Если мы используем метод итераций Ньютона Гаусса, то делаем линеаризацию нашей задачи около приближительного решения. Численное моделирование демонстрирует, что найденная матрица будет всегда особой для шести точек (т.е. с нулевым определителем). Таким образом, необходимо использовать по крайней мере семь точек, чтобы получить неособую линейную аппроксимацию. Обычно, много больше векторов используется, чтобы создать переопределённую систему, и, следовательно, снизить эффект шума. Ясно, существуют и вырожденные сценарии, в которых полученная система является особой, независимо от того каково число доступных характерных точек. Примеры таких сценариев включают полет выше полностью плоского или сферического ландшафта. Однако, в общем случае, где у ландшафта есть "интересная" структура, система является неособенной, и эти необходимые параметры могут быть получены.

Замечание 2: связь (18) нелинейна и, поэтому, аналитическое решение не существует. Таким образом, итерационная схема будет использоваться, чтобы решить эту систему. Устойчивый алгоритм, использующий итерации Ньютона и М-эстиматор будет описан в следующих разделах.

Замечание 3: Из замечание 2 можно видеть, что местоположение и сдвиг появляются линейно в полученных уравнениях. Используя псевдоинверсию, эти два вектора могут быть найдены в явной форме:

$$\begin{bmatrix} p_{12} \\ p_1 \end{bmatrix} = A_n^\dagger B_n, \quad (20)$$

и, после подстановки в (19):

$$(I - A_n A_n^\dagger) B_n = 0. \quad (21)$$

Это замечание приводит к следующим заключениям:

1. Если вращение известно с хорошей точностью, и шум измерения относительно низок, то положение и сдвиг могут быть найдены, путем решения линейного уравнения. Этот факт может быть важным, при сочетании процедуры, описанную здесь с другим измерением, например, с инерционной навигацией.

2. Уравнение (21) демонстрирует, что нахождение вращения (и абсолютного, и относительного) может быть отделено от нахождения местоположения/сдвига. Этот факт был также найден в [17], где оценивалась поза на основе ряда наземных ориентиров. В этой статье, так же в нашей, оценка получена, минимизируя целевую функцию, которая измеряет ошибки в *пространстве объекта*, а не на плоскости изображения (как в большинстве других работ). Это свойство позволяет разъединение задачи. Отметим однако, что в [17], разделены вращение и сдвиг, в то время как здесь 6 параметров абсолютного и относительного вращения отделены от 6 параметров местоположения камеры и сдвига.

3.2 Связи на основе эпиполярной геометрии.

Прежде, чем продолжить дальше, интересно проанализировать (15) в свете предыдущей работы по SFM и, в частности эпиполярной геометрии. Чтобы сделать это, следует получать основные соотношения в существующей структуре и системе обозначений. Напишем:

$${}^c_2 Q_T = \lambda_2 q_2 = p_{12} + \lambda_1 R_{12} q_1 \quad (22)$$

для некоторых скаляров и λ_1 (λ_2 см. Рис. 2).

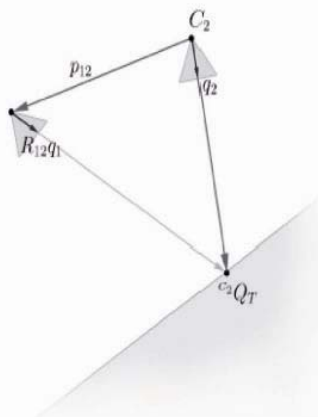


Рис. 2: Исследование сценария с точки зрения второй камеры (C_2). Перспективные проекции q_2 и q_1 определяются одной и той же особенностью ландшафта ${}^{c_2}Q_T$. Эпиполярная связь требует, чтобы два луча из соответствующих оптических центров (один в направлении q_2 , другой в направлении $p_{12} R_{12}q_1$) пересеклись.

Из этого следует, что:

$$p_{12} \times \lambda_2 q_2 = p_{12} \times \lambda_1 R_{12} q_1, \quad (23)$$

и следовательно:

$$q_2^T (p_{12} \times R_{12} q_1) = 0. \quad (24)$$

Для вектора, $x \in \mathbb{R}^3$ обозначим x^\wedge следующую кососимметрическую матрицу:

$$x^\wedge = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Тогда, векторное произведение между двумя векторами x и y может быть выражено как:

$$x \times y = x^\wedge y.$$

Используя эту систему обозначений, эпиполярная связь (24) может быть написана как:

$$q_2^T (R_{12} q_1)^\wedge p_{12} = 0 \quad (25)$$

и симметрично как:

$$q_1^T R_{12}^T q_2^\wedge p_{12} = 0 \quad (26)$$

Важное наблюдение здесь состоит в том что, если вектор p_{12} удовлетворяет вышеупомянутую связь, то вектор также $\kappa \cdot p_{12}$ также удовлетворяет эту связь, для любого числа κ . Это - двусмысленность, встроенная в задачу SFM. С другой стороны, связь (15) неоднородна и следовательно не страдает от этой двусмысленности. В терминах только сдвига (и только для одной характерной точки!), если p_{12} удовлетворяет (15) для данного R_1 и R_{12} , то также $p_{12} + \kappa q_2$ будет удовлетворять ему, и, следовательно, сдвиг эго-движения определен с точность до одномерного вектора. Однако, следующее выражение тождественно выполняется:

$$q_1^T R_{12}^T q_2^\wedge q_2 = 0, \quad (27)$$

и следовательно эпиполярная геометрия не обеспечивает дополнительное уравнение, которое позволило бы нам однозначно находить сдвиг. Кроме того, заметим, что (15) может быть написано, используя векторное произведение вместо оператора проекции как:

$$q_2^\wedge \left[p_{12} + \frac{R_{12} q_1 N^T}{N^T R_1 q_1} (Q_E - p_1) \right] = 0. \quad (28)$$

Принятие во внимание тождество

$$(R_{12} q_1)^T q_2^\wedge R_{12} q_1 \equiv 0, \quad (29)$$

возможно заключить, что (28) приводит к (26), и следовательно новая связь "содержит" классическую эпиполярную геометрию. Действительно, можно интерпретировать связь полученную в (15) как усиление эпиполярной геометрии. Она требует не только, чтобы два луча (в направлениях q_1 и q_2 из соответствующих оптических центров камер) пересеклись в одной точке, но, кроме того, чтобы эта точка пересечения лежала на касательной плоскости к поверхности, которая находится с помощью ЦКЛ. Наблюдение,

больше чем одной характерной точки позволило бы нам полностью вычислять сдвиг (по крайней мере, для заданных матриц вращения).

4. Навигационный алгоритм, основанный на компьютерном зрении, для коррекции инерциальной навигации с помощью фильтра Калмана

Основанные на компьютерном зрении навигационные алгоритмы были главной исследуемой проблемой в течение прошлых десятилетий. Алгоритм, используемый в этой статье, основан на геометрии многих изображений и карте местности. С помощью этого метода мы получаем позицию и ориентацию наблюдающей камеры. С другой стороны мы получаем те же самые данные из инерциальных навигационных методов. Чтобы скорректировать эти два результата, используется фильтр Калмана. Мы используем в этой статье расширенный фильтр Калмана для нелинейных уравнений [26].

Для инерциальных навигационных вычислений использовался Инерциальный Навигационный Системный Пакет для Matlab [27].

Вход фильтра Калмана состоит из двух частей. Первый - переменные X для уравнений движения. В нашем случае это - инерциальные навигационные уравнения. Вектор X состоит из пятнадцати компонентов: $[\delta x \ \delta y \ \delta z \ \delta V_x \ \delta V_y \ \delta V_z \ \delta \phi \ \delta \theta \ \delta \psi \ a_x \ a_y \ a_z \ b_x \ b_y \ b_z]$. Координаты $\delta x \ \delta y \ \delta z$ определены как разница между реальной позицией камеры и позицией, полученной из инерциального навигационного вычисления. Переменные $\delta V_x \ \delta V_y \ \delta V_z$ определены как разница между реальной скоростью камеры и скоростью, полученной из инерциального навигационного вычисления. Переменная $\delta \phi \ \delta \theta \ \delta \psi$ определена как углы Эйлера матрицы $D_r * D_c^T$. Здесь D_r матрица определена реальными углами Эйлера камеры относительно локальной системы координат (L-frame). С другой стороны D_c матрица определена углами Эйлера камеры относительно локальной системы координат (L-frame), полученными из инерциального навигационного вычисления. Необходимо обратить внимание, что найденные углы Эйлера $\delta \phi \ \delta \theta \ \delta \psi$ НЕ эквивалентны разнице между реальными углами Эйлера и углами Эйлера, полученными из инерциального навигационного вычисления. Однако для маленьких значений $\delta \phi \ \delta \theta \ \delta \psi$ поправки к этим углам могут быть добавлены линейно и таким образом эти углы могут использоваться в фильтре Калмана в случае маленьких ошибок. Такой выбор углов сделан, поскольку формулы, описывающие их эволюцию, намного более просты, чем формулы, описывающие эволюцию разницы между углами Эйлера. Переменные $a_x \ a_y \ a_z$ определены вектором смещения ускорения в инерциальных навигационных измерениях. Переменные $b_x \ b_y \ b_z$ определены вектором гироскопического смещения в инерциальных навигационных измерениях.

Второй вход фильтра Калмана - Z -результат измерений, основанных на навигационных алгоритмах компьютерного зрения. Вектор Z состоит из шести компонентов $[\delta x_m \ \delta y_m \ \delta z_m \ \delta \phi_m \ \delta \theta_m \ \delta \psi_m]$. Координаты $\delta x_m \ \delta y_m \ \delta z_m$ являются разницей между позицией камеры, измеренной на основе навигационного алгоритма компьютерного зрения, и позицией, полученной из инерциального навигационного вычисления. Переменная $\delta \theta_m \ \delta \psi_m$ определена как углы Эйлера матрицы $D_m * D_c^T$. Здесь D_m матрица определена углами Эйлера камеры относительно относительно локальной системы координат (L-frame), измеренных на основе навигационного алгоритма компьютерного зрения. С другой стороны D_c матрица определена углами Эйлера камеры относительно локальной системы координат (L-frame), полученными из инерциального навигационного

вычисления. Пусть переменная k определяет число шагов для дискретизации времени, используемых в фильтре Калмана.

Мы полагаем, что ошибки между значениями, полученными из инерциального навигационного вычисления, и реальными значениями линейно зависят от шума. Соответствующая ковариационная матрица шума обозначена Q_k . Диагональные элементы Q_k соответствуют скорости, определяются шумом ускорения и пропорциональны dt^2 : $Q_V \sim dt^2$, где dt интервал времени между t_k и t_{k-1} : $dt = t_k - t_{k-1}$. Диагональные элементы Q_k соответствуют углам Эйлера, определяются гироскопическим шумом и пропорциональны dt : $Q_A \sim dt$.

Мы предполагаем, что также ошибки между значениями, полученными на основе навигационного алгоритма компьютерного зрения, и реальными значениями линейно зависят от шума. Соответствующая ковариационная матрица шума обозначена R_k . Анализ ошибок, дающий эту матрицу, описан в [28].

Уравнения фильтра Калмана описывают как эволюцию оценок состояния X_k , описанных выше, так и эволюцию ковариационной матрицы P_k для переменных X_k .

Чтобы написать уравнения фильтра Калмана, мы должны определить еще две 15x15 матрицы: H_k и A_k . Матрица H_k является якобианом измерения, описывающая связь между предсказанным измерением $H_k * X_k$ и фактическим измерением Z_k , определенным выше. Диагональные элементы $H_k(1,1)$, $H_k(2,2)$, $H_k(3,3)$, описывающие координату и элементы $H_k(4,7)$, $H_k(5,8)$, $H_k(6,9)$, описывающие углы, равны единице. Остальная часть элементов равна нулю.

A_k матрица Якоби, описывающая эволюцию вектора X_k . Точное выражение для этой матрицы является очень сложным, поэтому мы используем приближенную формулу для A_k , пренебрегая эффектами кориолиса, вращением Земли и так далее. Позвольте $\phi \theta \psi$ быть углами Эйлера в L-frame, dV является вектором deltaV, полученным из инерциальных навигационных измерений, f_{vec} вектор ускорения в L-frame, DCM_{b-to-l} - матрица направляющих косинусов (для перевода из координатной системы летательного аппарата в L-frame).

Формулы определяющие A_k , следующие:

$$\Psi_{DCM} = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (48)$$

$$\Theta_{DCM} = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (49)$$

$$\Phi_{DCM} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (50)$$

$$DCM_{b-to-l} = \Phi_{DCM} \Theta_{DCM} \Psi_{DCM} \quad (51)$$

$$f_{vec} = DCM_{b-to-l} \frac{dV}{dt} \quad (52)$$

$$Phi(1:3,4:6) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (53)$$

$$Phi(4:6,7:9) = \begin{pmatrix} 0 & -f_{vec}(3) & f_{vec}(2) \\ f_{vec}(3) & 0 & -f_{vec}(1) \\ -f_{vec}(2) & f_{vec}(1) & 0 \end{pmatrix} \quad (54)$$

$$Phi(7:9,10:12) = -DCM_{b-to-l} \quad (55)$$

$$Phi(4:6,13:15) = -DCM_{b-to-l} \quad (56)$$

Остальные элементы матрицы Phi равны нулю.

$$A_k = I + Phi dt \quad (57)$$

Уравнения фильтра Калмана для временной эволюции следующие

$$X_k^- = [0000000000 a_{x_{k-1}} a_{y_{k-1}} a_{z_{k-1}} b_{x_{k-1}} b_{y_{k-1}} b_{z_{k-1}}] \quad (58)$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1} \quad (59)$$

Уравнения фильтра Калмана проектируют состояние и ковариационную матрицу с предыдущего временного шага $k-1$ на текущий временной шаг k .

Уравнения фильтра Калмана для измерения следующие:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (60)$$

$$X_k = X_k^- + K_k (Z_k - H_k X_k^-) \quad (61)$$

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (62)$$

Уравнения фильтра Калмана для измерения исправляют состояние и ковариационную матрицу в соответствии с измерением Z_k .

Найденный вектор X_k используется, чтобы обновить координаты, скорости, углы Эйлера, смещение ускорения и гироскопическое смещение для инерциальных навигационных вычислений на следующем шаге.

Численные расчеты были реализованы, чтобы исследовать эффективность фильтра Калмана и чтобы объединить эти два навигационных алгоритма. На Рис. 3, 4, 5 мы можем видеть, что для откорректированного пути ошибка координаты, полученная на основе двух навигационных методов с фильтрацией Калмана, намного меньше чем инерциальная навигационная ошибка координаты, полученная без фильтра Калмана. Улучшенные результаты с помощью фильтра Калмана были получены также для скорости, несмотря на то, что эта скорость не измеряется напрямую навигационным алгоритмом, использующим компьютерное зрение Рис. 6.

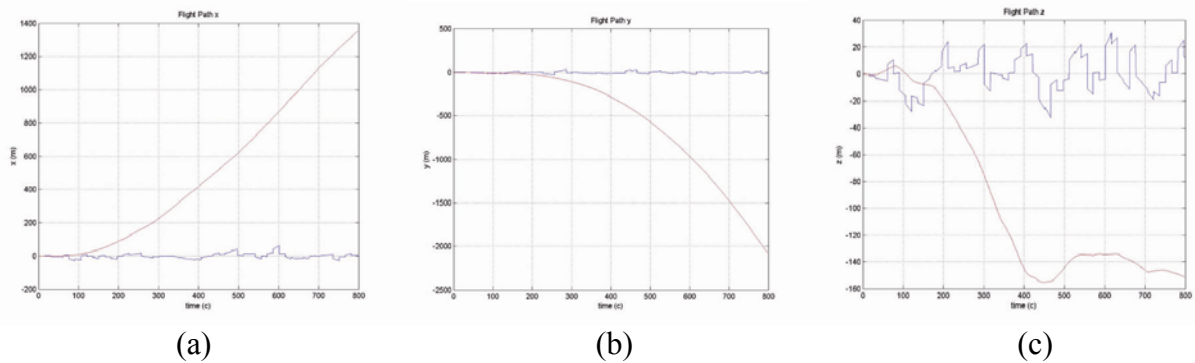


Рис. 3: Ошибки позиции ((a) для координаты x (b) для координаты y (c) для координаты z). Ошибки инерциального дрейфа отмечены красной линией, и ошибки, исправленные видео-навигацией, отмечены синей линией. Параметры: Высота 1000 м., FOV 60°, Число характерных точек 120, Разрешение 1000x1000, Расстояние между двумя положениями камер для проведения видеонаблюдения 200m, время между наблюдениями 15 сек

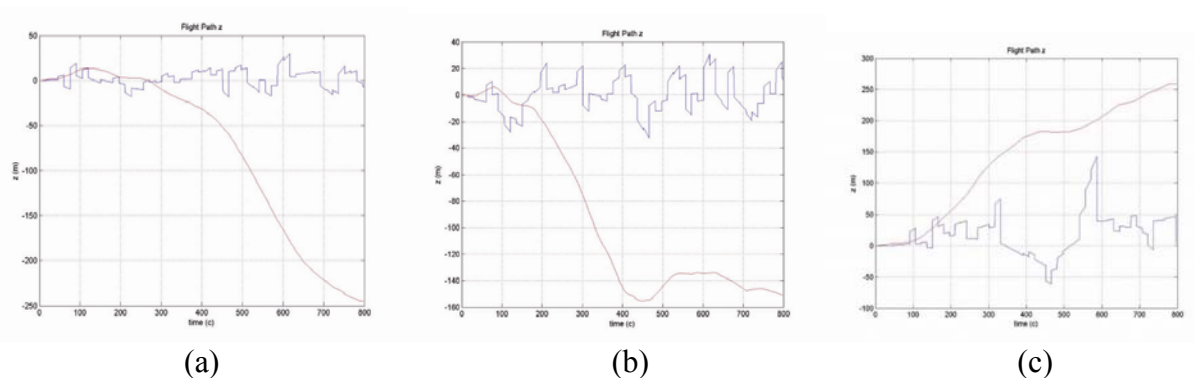


Рис. 4: Ошибки позиции x, y, z для координат инерциального дрейфа отмечены красной линией, и ошибки исправленного пути отмечены с синей линией. Параметры: FOV 60°, Число характерных точек 120, Разрешение 1000x1000, Расстояние между двумя

положениями камер для прведения видеонаблюдения =200m, время между наблюдениями 15 сек, высота а) 700 м. б) 1000 м. с) 3000 м.

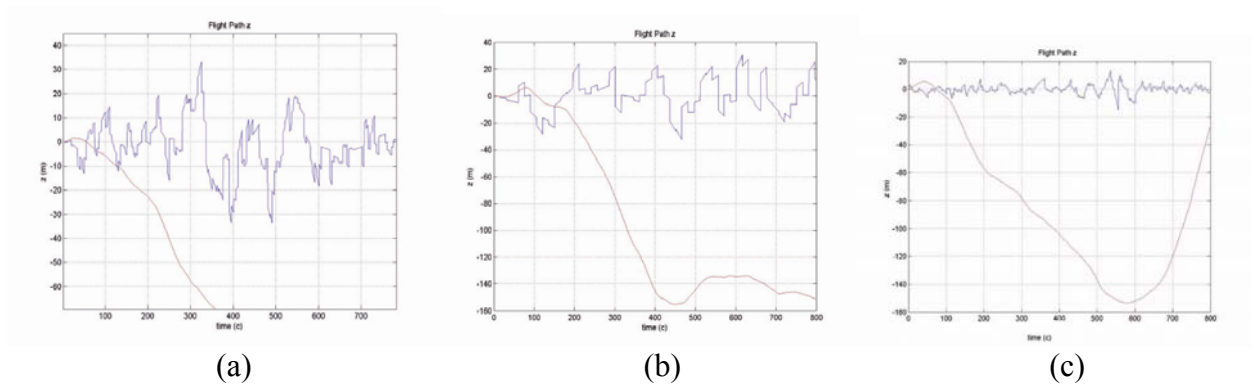


Рис. 5: Ошибки положения для x, y, z координат инерциального дрейфа отмечены с красной линией, и ошибки исправленного пути отмечены с синей линией. Параметры: FOV 60°, Число характерных точек 120, Расстояние между двумя положениями камер для прведения видеонаблюдения =200m, время между наблюдениями 15 сек, Высота 1000 м., Разрешение а) 500x500 б) 1000x1000 с) 4000x4000

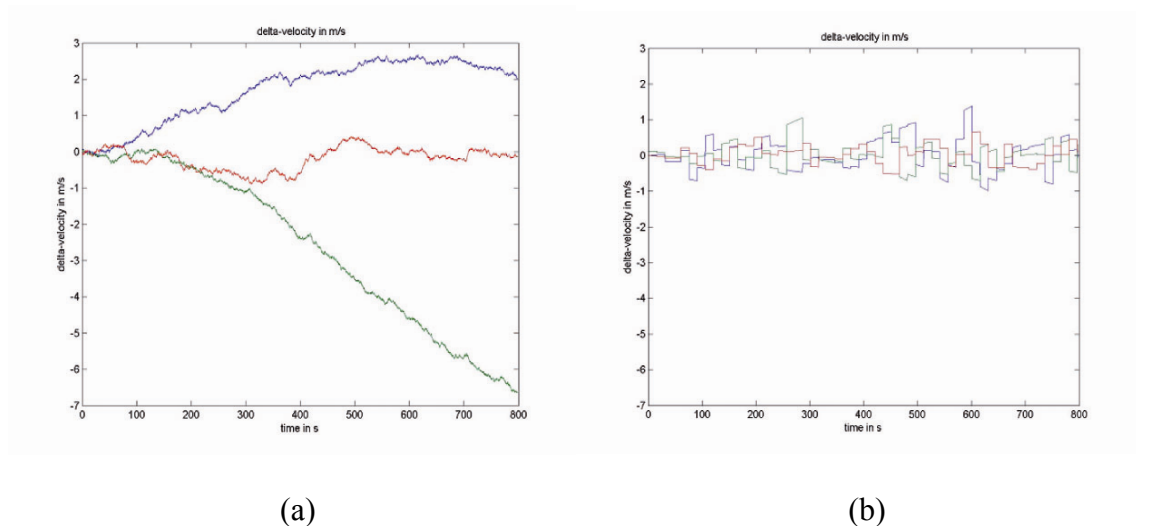


Рис. 6: (а) Ошибки скорости для инерциального дрейфа (x y z компоненты), и (б) ошибки скорости, исправленные видео-навигацией (x y z компоненты). Высота Параметров 1000 м., FOV 60°, Число характерных точек 120, Разрешение 1000x1000, Расстояние между двумя положениями камер для прведения видеонаблюдения =200m, время между наблюдениями 15 сек.

5. Анализ погрешностей

Остальная часть этой работы имеет дело с анализом погрешностей предложенного алгоритма. Чтобы оценить работу алгоритма, целевая функция процесса минимизации должна быть сначала определена: для каждого из n векторов оптического потока определена функция $f_i : \mathbb{R}^{12} \rightarrow \mathbb{R}^3$, которая является левой частью уравнения (16):

$$f_i(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) =$$

$$= \mathcal{P}(q_2, q_2) \left[p_{12} + R_{12} L_i (G_{E_i} - p_1) \right] / |{}^{c_2} G| \quad (45)$$

В вышеупомянутом выражении, R_{12} и L_i являются функциями переменных $(\phi_{12}, \theta_{12}, \psi_{12})$ и $(\phi_1, \theta_1, \psi_1)$ соответственно. Дополнительно, функция $F: \mathbb{R}^{12} \rightarrow \mathbb{R}^{3n}$ будет определена как объединение функций $f_i: F(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) = [f_1, \dots, f_n]^T$. Согласно этой системе обозначений, цель алгоритма состоит в том, чтобы найти двенадцать параметров, которые минимизируют $M(\theta, D) = \|F(\theta, D)\|^2$, где θ представляет собой вектор из 12 параметров, которые должны быть оценены, а объединение D всех данных, которые получают из оптического потока и DTM. Если было бы D свободно от ошибок, истинные параметры были получены. Так как D содержит некоторое ошибочное возмущение, оцененные параметры дрейфуют к ошибочным значениям. В [10] показано, что связь между неопределенностью в данных и неопределенностью в оцениваемых параметрах может быть описана следующей аппроксимацией первого порядка:

$$\Sigma_\theta = \left(\frac{dg}{d\theta} \right)^{-1} \left(\frac{dg}{dD} \right) \Sigma_D \left(\frac{dg}{dD} \right)^T \left(\frac{dg}{d\theta} \right)^{-1} \quad (46)$$

Здесь, Σ_θ и Σ_D - матрицы ковариантности параметров и данных, соответственно. g определено следующим образом:

$$g(\theta, D) = \frac{d}{d\theta} M(\theta, D) = \frac{d}{d\theta} F^T F = 2J_\theta^T F \quad (47)$$

$J_\theta = dF/d\theta$ - $(3n \times 12)$ Якобиан матрицы F относительно этих двенадцати параметров.

Игнорируя элементы второго порядка, производные g могут быть определены:

$$\frac{dg}{d\theta} \approx 2J_\theta^T J_\theta \quad (48)$$

$$\frac{dg}{dD} \approx 2J_\theta^T J_D \quad (49)$$

$J_D = dF/dD$ - $(3n \times m)$ Якобиан матрицы F относительно m компонентов данных.

Подставляя (48) и (49) в (46), приходим к следующему выражению:

$$J_T = (J_\theta^T J_\theta)^{-1} J_\theta^T$$

$$\Sigma_\theta = J_T \cdot (J_D \Sigma_D J_D^T) \cdot J_T^T \quad (50)$$

Центральный компонент $J_D \Sigma_D J_D^T$ представляет неопределенность F , в то время как псевдообратная матрица $(J_\theta^T J_\theta)^{-1} J_\theta^T$ преобразует неопределенность в F к неопределенности этих двенадцати параметров. В следующих подразделах, J_θ , J_D и Σ_D явно получены.

5.1 Вычисление J_θ

Простое дифференцирование f_i , представленного в (45), приведет к следующим результатам:

$$N_p(q_2, {}^{c_2} G) = \mathcal{P}(q_2, q_2) \mathcal{P}({}^{c_2} G, {}^{c_2} G) / |{}^{c_2} G| \quad (51)$$

$$\frac{df}{dp_1} = -N_P(q_2, {}^{c_2}G)R_{12}\mathcal{L} \quad (52)$$

$$\frac{df}{d\alpha_1} = -N_P(q_2, {}^{c_2}G)R_{12}\mathcal{L}\left(\frac{d}{d\alpha_1}R_1\right)\mathcal{L}(G_E - p_1) \quad (53)$$

$$\frac{df}{dp_{12}} = N_P(q_2, {}^{c_2}G) \quad (54)$$

$$\frac{df}{d\alpha_{12}} = N_P(q_2, {}^{c_2}G)\left(\frac{d}{d\alpha_{12}}R_{12}\right)\mathcal{L}(G_E - p_1) \quad (55)$$

В выражениях (53) и (55): $\alpha_1 = \phi_1, \theta_1, \psi_1$ и: $\alpha_{12} = \phi_{12}, \theta_{12}, \psi_{12}$. Якобиан получен J_θ простым объединением вышеупомянутых дифференцирований.

5.2 Вычисление J_D

Перед вычислением J_D должен быть явно определен вектор данных D . Два типа данных используются предложенным навигационным алгоритмом: данные, полученные из оптического потока, и данные из ЦКЛ. Каждый вектор потока начинается в q_1 и заканчивается в q_2 . Можно рассмотреть q_1 's как проекцию некоторой произвольно выбранной характерной точки местности, в то время как q_2 представляет собой новую проекцию той же самой характерной точки на втором снимке. Таким образом, ошибки потока реализуются через векторы q_2 .

Ошибки ЦКЛ влияют на G_E и N векторов в уравнении связи. Как и прежде, линеаризация ЦКЛ будет использоваться. Для простоты, ориентация локальной линеаризации ландшафта, выраженная нормалью, рассматривается как правильная, в то время как высота соответствующей плоскости могла бы быть ошибочной. Связь между ошибкой высоты и ошибкой будет G_E получена в следующем подразделе. Следуя из вышеупомянутого, все q_1 и N могут быть влечены из вектора данных D . Это может быть определено как объединение весь q_2 , сопровождаемое объединением всех G_E .

Векторы данных i -ой характерной точки q_{2_i} и G_{E_i} появляются только в уравнениях для i -ой точки. Таким образом, полученная матрица Якоби - $J_D = [J_q, J_G]$ является объединением двух блок-диагональных матриц: J_q и поледующей J_G . i 'th диагональный блокочный элемент - это 3×3 матрицы df_i/dq_{2_i} и df_i/dG_{E_i} для J_q и J_G соответственно:

$$\frac{df}{dq_2} = \frac{-1}{\|q_2\|^2} \left[\left(q_2^T \cdot {}^{c_2}G \right) I + q_2 \cdot {}^{c_2}G^T \right] P(q_2, q_2) / \left| {}^{c_2}G \right| \quad (56)$$

$$\frac{df}{dG_E} = N_P(q_2, {}^{c_2}G)R_{12}\mathcal{L} \quad (57)$$

${}^{c_2}G$ в выражении (56) - характерная точка местности G снимка второй камеры, как определено в (11).

5.3 Вычисление Σ_D

Как упоминалось выше, вектор данных D построен из объединения всех q_2 , сопровождаемого объединением всех G_E . Таким образом, Σ_D должна описывать неопределенность в этих элементах. Так как q_2 и G_E получены из двух различных, и некоррелированных процессов, ковариантность, связывающая их, будет ноль. Это приводит к двум блок-диагональным матрицам:

$$\Sigma_D = \begin{bmatrix} \Sigma_q & 0 \\ 0 & \Sigma_G \end{bmatrix} \quad (58)$$

В этой работе ошибки положения изображения и высоты, определяемой ЦКЛ, предполагаются аддитивными, описываемые Гауссовым распределением с нулевым средним и стандартным отклонением σ_l и σ_h , соответственно. Каждый вектор q_2 - проекция на плоскости изображения, где фокусное расстояние принимается равным единице. Следовательно, нет никакой неопределенности о - z компонент. Так как, ради простоты, нормальное изотропное распределение принимается, корреляционная матрица измерения изображения определена следующим образом:

$$\Sigma_{q_i} = \sigma_l^2 \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \quad (59)$$

а матрица Σ_q - это матрица с матрицами Σ_{q_i} вдоль ее диагонали.

В [11] изучена точность высоты точки на местности, полученной интерполяцией соседних точек из сетки, определяемой ЦКЛ. Показано, что зависимость между этой точностью и местоположением, для которого интерполируется высота является незначительной. Здесь, вышеупомянутое наблюдение было принято во внимание, и постоянное стандартное отклонение было установлено для всех измерений высот ЦКЛ. Хотя и существует зависимость между неопределенностью точек G_E , эта зависимость будет проигнорирована в следующих дифференцированиях ради простоты. Таким образом, блок-диагональная матрица, полученная для Σ_G , содержащую 3×3 корреляционные матрицы Σ_{G_i} вдоль ее диагонали, которая будет получена следующим образом. Рассмотрим луч, посланный из точки p_1 вдоль направления R_1q_1 . Этот луч должен был пересечь ландшафт в точке $G_E = p_1 + \lambda R_1q_1$ для некоторого λ . Но из-за ошибки высоты ЦКЛ, другая точка $\tilde{G}_E = (\tilde{x}, \tilde{y}, \tilde{h})^T$ была получена. Пусть h является истинной высотой ландшафта в точке (\tilde{x}, \tilde{y}) , а $H = (\tilde{x}, \tilde{y}, h)$ является соответствующей трехмерной точкой на ландшафте.

Используя то, что H принадлежит истинной касательной плоскости к ландшафту, получаем:

$$N^T(G_E - H) = N^T(p_1 + \lambda R_1q_1 - H) = 0 \quad (60)$$

Извлекая λ из (60) и подставляя это назад в уравнение для G_E получаем выражение:

$$G_E = p_1 + R_1 \mathcal{L}(H - p_1) \quad (61)$$

Для вычисление неопределенности G_E должна быть найдена производная G_E относительно h :

$$\frac{dG_E}{dh} = R_1 \mathcal{L} \cdot (0 \ 0 \ 1)^T = \frac{R_1 q_1}{N^T R_1 q_1} \quad (62)$$

Вышеупомянутый результат был получен, используя факт, что z компонент вектора N равен 1: $N = (-\nabla DTM \ 1)^T$. Наконец, неопределенность G_E выражается следующей корреляционной матрицей:

$$\Sigma_{G_i} = \left(\frac{dG_E}{dh} \right) \cdot \sigma_h^2 \cdot \left(\frac{dG_E}{dh} \right)^T = \sigma_h^2 \cdot \frac{R_1 q_1 q_1^T R_1^T}{(N^T R_1 q_1)^2} \quad (63)$$

5.4 Вычисление Σ_{C_2}

Алгоритм, представленный в этой работе, оценивает позицию и ориентацию при первом положении камеры и эго-движение. Обычно, самые интересные параметры при навигации - это второе положение камеры. Оно отражают наиболее новую информацию о местоположении камеры. Вторая позиция и ориентация камеры могут быть получены напрямую как композиция первой позиции и ориентации и эго-движения камеры:

$$p_2 = p_1 - R_1 R_{12}^T p_{12} \quad (64)$$

$$R_2 = R_1 R_{12}^T \quad (65)$$

Неопределенность вторых позиции и ориентации будет описана 6×6 корреляционной матрицей которая может быть получена из уже полученной 12×12 матрицы ковариантности 12×12 умножением Σ_θ с обеих сторон на J_{C_2} . Последний символ обозначает Якобиан упомянутых выше шести параметров, описывающих C_2 , относительно этих двенадцати параметров. С этой целью, три Эйлеровых угла, ϕ_2 , θ_2 и ψ_2 должны быть извлечены из (65), используя следующие уравнения:

$$\phi_2 = \arctan\left(\frac{R_2(2,3)}{R_2(3,3)}\right) \quad (66)$$

$$\theta_2 = \arcsin(-R_2(1,3)) \quad (67)$$

$$\psi_2 = \arctan\left(\frac{R_2(1,2)}{R_2(1,1)}\right) \quad (68)$$

Простое дифференцирование, а затем и объединение вышеупомянутых выражений приводит к необходимому Якобиану, который используется, чтобы распространить неопределенность от C_1 и эго-движения к C_2 . Найденная корреляционная матрица - Σ_{C_2} та же самая, что и корреляционная матрица измерения R_k , описанная в разделе о фильтре Кальмана.

$$R_k = \Sigma_{C_2} \quad (69)$$

6. Расходимость метода. Необходимые пороговые значения для сходимости метода.

В предыдущем Разделе мы рассматривали Анализ погрешностей для видео навигационного метода. Но его рассмотрение правильно, только если первое грубое приближенное решение близко к истинному. Если это не верно, нелинейные эффекты, могут появиться, или даже мы можем найти неправильный локальный минимум. В этом случае метод может начать расходиться. Мы можем получить такой результат:

1) если большое количество характерных точек, найденных с грубыми ошибками в их положении на местности появляется

2) если ситуация близка к вырожденной. В этом случае ошибки положения или ориентации являются слишком большими. Это может случиться например для малого числа характерных точек на местности, плоского ландшафта, небольшого угла зрения камеры.

3) если начальное положение и ориентация для процесса итераций слишком далеки от истинных значений

В следующих подразделах мы рассматриваем пороговые условия, которые позволяют нам избегать таких ситуаций.

Если в некотором случае даже одно из этих пороговых условий детектирует ошибку, мы не используем для этого случая визуальную коррекцию и используем только обычный результат инерциальной навигации. Если такая ситуация повторяется три раза подряд мы прекращаем использовать визуальный навигационный метод вообще и не используем его коррекцию также для последнего правильного случая. Теперь позвольте нам обсудить вышеописанные три случая в деталях.

6.1 Обработка ситуации с большим количеством характерных точек, найденных с грубыми ошибками

Чтобы работать с реальными данными, процедура для обработки большого числа характерных точек должна быть включена в алгоритм. Цель существующего раздела состоит в том, чтобы описать текущую процедуру, которая, кажется, работает удовлетворительно на практике. Три вида грубых ошибок нужно рассмотреть:

1. Ошибки связаны с неверным нахождением соответствия между характерными точками.

2. Ошибки, вызванные формой ландшафта, и

3. Ошибки, вызванные относительно большим несоответствием между имеющейся ЦКЛ и наблюдаемым ландшафтом.

Последние два вида ошибок проиллюстрированы на Рис. 7. Ошибки, вызванные формой ландшафта, появляются для особенностей ландшафта, расположенных близко к большим вариациям глубины. Например, рассмотрите два холма, один ближе к камере, другой дальше, и характерная точка ландшафта Q расположена на более близком холме. Алгоритм, прослеживающий луч, используя ошибочную ориентацию может "пропустить" ближайший холм и ошибочно поместить особенность в дистальный. Само собой разумеется, ошибка между истинными и оцененными местоположениями не покрывается линеаризацией. Чтобы визуально представить себе ошибки, связанные с относительно большим рассогласованием между ландшафтом и ЦКЛ, предположим, что здание присутствовало на ландшафте, когда ЦКЛ был приобретен, но больше его там нет, когда эксперимент имеет место. Алгоритм, прослеживающий луч, определит местонахождение особенности на здании, хотя истинная особенность ландшафта принадлежит фону, который мы теперь видим.

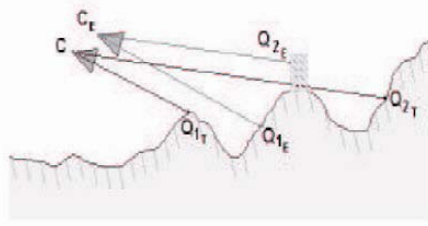


Рис. 7: Ошибки, вызванные формой ландшафта и рассогласованием с ЦКЛ. Здесь C_T - истинное, а C_E оценочное положение и ориентация камеры, соответственно. и Q_{1E}, Q_{2E} грубые ошибки вызванные формой ландшафта или рассогласованием ландшафт/ЦКЛ, соответственно.

Как обсуждено выше, учет многих характерных точек производится в смысле наименьших квадратов для переменных движения и позы. Учитывая чувствительность наименьших квадратов к неправильным данным, включение одной или более грубой ошибки может привести к расходимости к неправильному решению. Возможный способ обойти эту трудность состоит в использовании М-эстиматора, в котором оригинальное решение заменено взвешенной версией. В этой версии небольшой вес приписывается точкам, имеющих большие ошибки, таким образом, минимизируя их влияние на решение. Более одетально, рассмотрим функцию $f_i(\Theta)$ определенную в (45) и вытекающей из i -th пары соответствий. В отсутствии шума эта функция должна быть равна нулю при истинной позе и эго-движении и, следовательно, используя стандартную систему обозначений, можно определить разность $r_i(\Theta) = \|f_i(\Theta)\|$. Используя М-эстиматор, решение для Θ (это двенадцать параметров, которые должны быть оценены), получается, используя итерационную схему для наименьших квадратов, с пересчитывающимися весами:

$$\Theta = \operatorname{argmin} \sum_{i=1}^n w_i r_i^2. \quad (70)$$

Веса w_i повторно вычисляются заново после того, как каждая итерация обновила разности. В нашем выполнении мы использовали так называемую функцию *Geman-McClure*, для которой веса даются:

$$w(x) = \frac{1}{(1+x^2)^2}. \quad (71)$$

Расчетные веса используются, чтобы построить взвешенную псевдообратную матрицу, которая заменяет регулярную псевдоинверсию, появляющуюся для J_T в (50). См. [18] для дальнейших деталей о М-эстиматоре. Позвольте нам определять матрицу весов W , который позволяет нам уменьшать влияние выбросов

$$r_i = \|f_i(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12})\|$$

$$medR = median(x_i)$$

$$R_i = w(r_i/medR) \quad (72)$$

где $i = 1, \dots, n$ номер характерной точки.

Матрица весов W ($3n \times 3n$) может быть найдена, следующим образом: для диагональных элементов W мы можем написать: $W_{ii} = R_k$, где k - это целая часть числа $[(i-1)/3+1]$. Недиагональные элементы для $W_{ij} = 0$ $i \neq j$.

Вместо уравнения (50) мы используем новое:

$$JT = (J_\theta^T W J_\theta)^{-1} J_\theta^T W$$

$$\Sigma_\theta = JT \cdot (J_D \Sigma_D J_D^T) \cdot JT^T \quad (73)$$

Если мы знаем два положения камеры и положение характерной точки на первой фотографии, мы можем найти положение характерной точки на второй фотографии. Если расстояние между истинным положением некоторой соответствующей особенности на второй фотографии и положением, найденным ранее описанным методом, больше чем, $3\sigma_I$, мы рассматриваем такую характерную точку как грубую ошибку. Давайте определим как N_i число грубых ошибок при начальной аппроксимации положения камер и ориентаций (то есть сделанную до использования визуальной навигации) и как N_f число грубых ошибок после использования визуальной навигации. Следующие условия позволяют нам избежать слишком большого количества грубых ошибок:

$$N_i \geq N_f$$

$$\frac{N_f}{N} < threshold_{\%} \quad (74)$$

где N - полное число грубых ошибок, а $threshold_{\%}$ это некоторое пороговое значение. Мы выбрали его равным 0.1.

6.2 Вырожденный случай больших ошибок.

Для вырожденного случая матрица в $J_\theta^T W J_\theta$ уравнении (73) может быть вырожденной. Это определяется следующим пороговым условием:

$$rcond(J_\theta^T W J_\theta) > threshold_{rcond} \quad (75)$$

где $rcond()$ - Matlab функция для нахождения оценки вырожденности матрицы. ($0 < rcond() < 1$). Пороговое значение выбрано $threshold_{rcond} = 10^{-16}$.

Вырожденный случай из-за небольшого количества характерных точек, плоского ландшафта или небольшого угла зрения камеры дается следующим пороговым условием:

$$\frac{\sqrt{[\Sigma_{C_2}]_{ii}}}{(3\sigma_I/f)h} < threshold_{dist} \quad (76)$$

где $i = x, y, z$ - это координатные индексы для диагональных элементов корреляционной матрицы Σ_{C_2} . Здесь $f = 1$ фокусное расстояние камеры, h - высота положения камеры. Выражение $\frac{3\sigma_1}{f}h$ дает нам максимальное изменение положения камеры, позволяющее ошибке положения характерной точки на фотографии оставаться меньшим, чем размер пиксела. Пороговое значение $threshold_{dist} = 40$.

$$3\sqrt{[\Sigma_{C_2}]_{ii}} < L_{ground-dist} \quad (77)$$

где координатны $i = x, y, z$ - это координатные индексы для диагональных элементов корреляционной матрицы Σ_{C_2} . $L_{ground-dist}$ - характерная длина, на которой рельеф сильно меняется

$$\frac{\sqrt{[\Sigma_{C_2}]_{ii}}}{(3\sigma_1/f)} < threshold_{angle} \quad (78)$$

где $i = \phi, \theta, \psi$ угловые индексы для диагональных элементов корреляционной матрицы Σ_{C_2} . Выражение $\frac{3\sigma_1}{f}h$ дает нам максимальное изменение положения камеры, позволяющее ошибке положения характерной точки на фотографии оставаться меньшим, чем размер пиксела. Пороговое значение $threshold_{angle} = 40$.

$$3\sqrt{[\Sigma_{C_2}]_{ii}} < \frac{L_{ground-dist}}{h} \quad (79)$$

где $i = \phi, \theta, \psi$ -угловые индексы для диагональных элементов корреляционной матрицы Σ_{C_2} .

Вырожденный случай из-за небольших данных (малое расстояние между двумя положениями камеры, используемыми в видео-навигационном методе), дает следующие пороговые условия:

$$\frac{\sqrt{[\Sigma_\theta]_{ii}}}{\|p_{12}\|} < threshold_{dist_{12}} \quad (80)$$

где $i = x_{12}, y_{12}, z_{12}$ - взаимные координатные индексы для диагональных элементов корреляционной матрицы Σ_θ . Пороговое значение $threshold_{dist_{12}} = 0.1$.

$$\frac{\sqrt{[\Sigma_\theta]_{ii}}}{(\|p_{12}\|/h)} < threshold_{angle_{12}} \quad (81)$$

где $i = \phi_{12}, \theta_{12}, \psi_{12}$ - взаимные угловые индексы для диагональных элементов корреляционной матрицы Σ_θ . Пороговое значение $threshold_{angle_{12}} = 0.1$.

6.3 Начальное положение камеры слишком далеко от ее истинного или итогового расчетного положения или ориентации.

Позвольте нам определять пороговые условия, чтобы избежать начального положения камеры, которое слишком далеко от ее истинного положения или ориентации. Корреляционная матрица P_k^- , полученная с помощью инерциальной навигации и предыдущих коррекций инерциальной навигации с помощью видео-навигации фильтром Калмана, описана в разделе о фильтре Калмана.

$$3\sqrt{[P_k^-]_{ii}} < L_{ground-dist} \quad (82)$$

где $i = x, y, z$ - это координатные индексы для диагональных элементов корреляционной матрицы P_k^- .

$$3\sqrt{[P_k^-]_{ii}} < \frac{L_{ground-dist}}{h} \quad (83)$$

где $i = \phi, \theta, \psi$ -угловые индексы для диагональных элементов корреляционной матрицы P_k^- .

Позвольте нам определять пороговые условия, позволяющие избежать начального состояния камеры слишком далекого от ее конечного состояния. Следующие четыре уравнения дают нам, разность между начальным и конечным состоянием, получающиеся как коррекция инерциальной навигации видео-навигационным методом с помощью фильтра Калмана:

$$\delta p_2 = |p_{2_final} - p_{2_init}| \quad (84)$$

$$\delta p_{12} = |p_{12_final} - p_{12_init}| \quad (85)$$

$$\delta \alpha_2 = |\alpha_{2_final} - \alpha_{2_init}| \bmod(2\pi) \quad (86)$$

$$\delta \alpha_{12} = |\alpha_{12_final} - \alpha_{12_init}| \bmod(2\pi) \quad (87)$$

$$3(\sqrt{[P_k^-]_{ii}} + \sqrt{[\Sigma_{C_2}]_{ii}}) > \delta p_{2i} \quad (88)$$

где $i = x, y, z$ - это координатные индексы для диагональных элементов корреляционных матриц P_k^- и Σ_{C_2}

$$3(\sqrt{[P_k^-]_{ii}} + \sqrt{[\Sigma_{C_2}]_{ii}}) > \delta \alpha_{2i} \quad (89)$$

где $i = \phi, \theta, \psi$ - угловые индексы для диагональных элементов корреляционных матриц P_k^- и Σ_{C_2}

$$\frac{\delta p_{12i}}{\|p_{12}\|} < threshold_{dist_{12}} \quad (90)$$

где $i = x_{12}, y_{12}, z_{12}$ - взаимные координатные индексы.

$$\frac{\delta \alpha_{12i}}{(\|p_{12}\|/h)} < threshold_{angle_{12}} \quad (91)$$

где $i = \phi_{12}, \theta_{12}, \psi_{12}$ - взаимные угловые индексы.

7 Результаты Моделирования

7.1 Зависимость анализа погрешностей от различных факторов.

Цель следующего раздела состоит в том, чтобы изучить влияние различных факторов на точность оценок, предложенных алгоритмом. Уравнение, описанные в предыдущих разделах, используется, чтобы определить неопределенность оценок для множества моделируемых сценариев. Каждый проверенный сценарий охарактеризован следующими параметрами: число характерных точек оптического потока, используемых алгоритмом, разрешение изображения, сеточным интервалом ЦКЛ (разрешение ЦКЛ), амплитуда холмов/гор на наблюдаемом ландшафте и величина компонентов эго-движения. При каждом моделировании все параметры кроме исследованного установлены согласно предопределенному множеству параметров. В этом *сценарии по умолчанию* используется камера с 400×400 разрешением. Она летит на высоте 500 м выше ландшафта. Размер ландшафта - 3×3 км с разностями возвышения на 300 м (Рис. 13 (b)). ЦКЛ с шагом сетки 30 м используется, чтобы смоделировать ландшафт (Рис. 10 (c)). Разрешение ЦКЛ приводит к стандартному отклонению 2.34 м. для измерений высоты. Сценарий по умолчанию также определяет число характерных точек оптического потока как приблизительно 170, где эго-движение $\|p_{12}\| = 40m$ и $\|(\phi_{12}, \theta_{12}, \psi_{12})\| = 10^\circ$, используемые для вычисления оптического потока. Каждое из моделирований описанных ниже исследует влияние различных параметров по отдельности. Множество значений исследовано, и 150 случайных испытаний выполнены для каждого проверенного значения. Для каждого испытания положение камеры и ориентация были выбраны беспорядочно, кроме высоты камеры, которая определена параметрами сценария. Дополнительно, направления сдвига и вращение для эго-движения были сначала выбраны наугад и затем нормализованы к требующейся величине.

На Рис. 8 представлены первые результаты моделирования. При этом моделировании число характерных точек оптического потока, которые используются алгоритмом, различно и его влияние на полученную точность, C_2 и эго-движение изучено. Все параметры были установлены на их значения по умолчанию за исключением числа характерных точек. Рис. 8 (a) представляет стандартные отклонения второго снимка камеры, в то время как отклонения эго-движения показаны на Рис. 8 (b). Как ожидается, точность улучшается при увеличении числа характерных точек, хотя уточнение

становится незначительным после того, как число характерных точек достигает приблизительно 150.

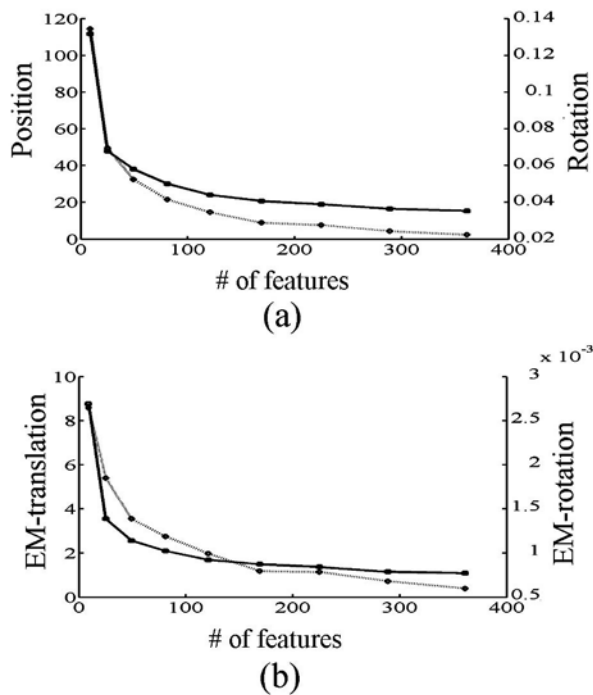
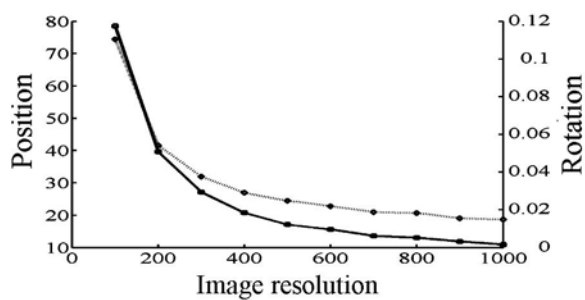
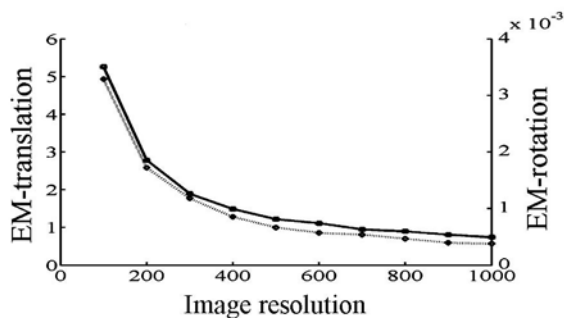


Рис. 8: Среднее стандартное отклонение второго положения и ориентации (а), это - движения и вращение (b) относительно числа характерных точек оптического потока. В обеих диаграммах левая вертикальная ось измеряет поступательные отклонение (в метрах) и соответствует сплошной линии диаграммы, в то время как правая вертикальная ось измеряет вращательные отклонения (в радианах) и соответствует пунктирной линии диаграммы

При втором моделировании влияние разрешения изображения было изучено (Рис. 9). Предполагалось, что измерения содержат неопределенность в полупиксель, где размер пикселя нам диктует разрешение изображения. Очевидно, точность улучшается при увеличении разрешения изображения, так как качество данных оптического потока, непосредственно зависит от этого параметра.



(a)



(b)

Рис. 9: Среднеквадратическое отклонение для второго положения и ориентации камеры (а), и для эго-движения (сдвиг и вращение) (b) приведено для различных разрешений изображения

Влияние шага сетки ЦКЛ - цель следующего моделирования. Различные разрешения ЦКЛ были проверены. Они изменялись от 10 м до чрезвычайно грубого разрешения 190 м между смежными точками сетки (см. Рис. 10). Обратим внимание, что полученная точность, изменяется линейно при изменении шага сетки ЦКЛ (см. Рис. 11). Это явление может быть понято так. Как было объяснено в предыдущем разделе, разрешение ЦКЛ не затрагивает точность непосредственно, а скорее это влияет на неопределенность высоты, которая включена в вычисление точности. Как может быть замечен в Рис. 12, среднеквадратическое отклонение высот ЦКЛ увеличивается линейно с шагом сетки ЦКЛ, что является причиной для полученных результатов.

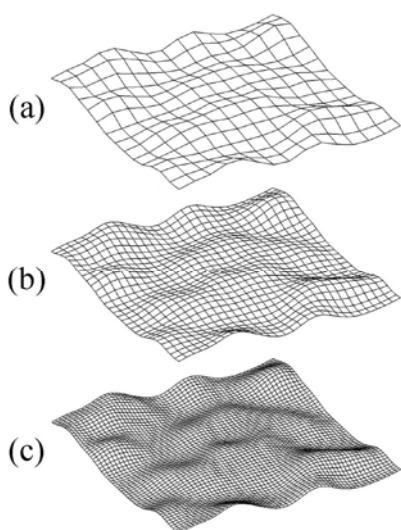
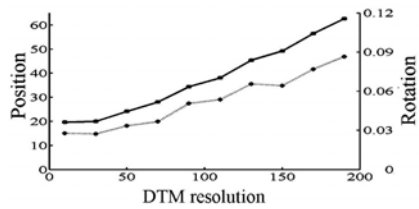
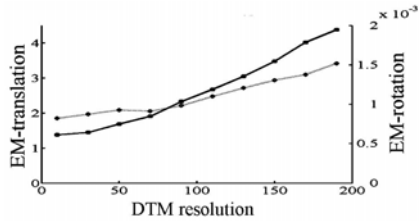


Рис. 10: Различные разрешения DTM: (а) шаг сетки = 190 м., (b) шаг сетки = 100 м., (с) шаг сетки = 30 м.



(a)



(b)

Рис. 11: Среднеквадратическое отклонение для второго положения и ориентации камеры (а), и для эго-движения (сдвиг и вращение) (b) приведено для различных шагов сетки ЦКЛ

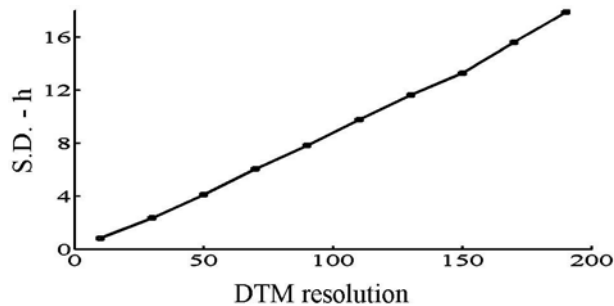


Рис. 12: Среднеквадратическое отклонение измерения высоты ЦКЛ относительно шага сетки ЦКЛ

Другое моделирование демонстрирует значимость структуры ландшафта для точности оценок. В крайнем сценарии полета над плоским ландшафтом наблюдаемые характерные точки ландшафта не содержат необходимую информацию для расчета положений камеры, и вырожденная система будет получена. С увеличением разности высот и изменчивости ландшафта, особенности становятся более информативными, и лучшие оценки могут быть получены. Для этого моделирования изменчивости ландшафта ЦКЛ были определены масштаб, который менялся от 50 м. до 450 м. (Рис. 13). Подчеркнем, что, в то время как структура ландшафта играет решающую роль при оценке положения и ориентации камеры вместе с поступательным компонентом эго-движения, у него нет никакого прямого действия на вращательный компонент эго-движения. Поскольку оптический поток - композиция двух векторных полей - сдвиг и вращение, информация о вращательном эго-движении содержится только во вращательном компоненте оптического потока. Так как высоты характерных точек влияют только на поступательный компонент потока, ожидается, что у переменных разностей высоты или любого другого структурного изменения в ландшафте не будет никакого воздействия на оценку вращения эго-движения. Вышеупомянутые особенности хорошо продемонстрированы на Рис. 14.

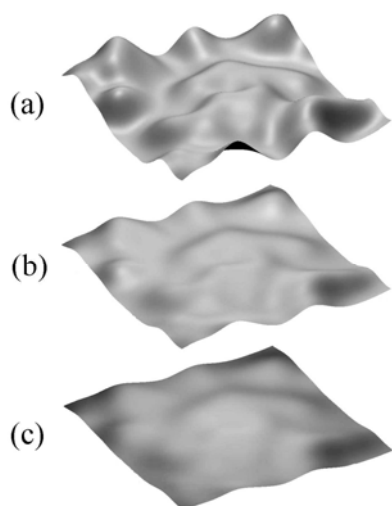


Рис. 13: Разности возвышений ЦКЛ: (a) 150 м., (b) 300 м., (c) 450 м.

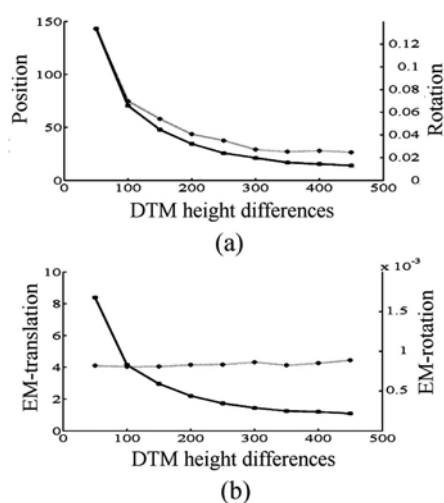


Рис. 14: Среднеквадратическое отклонение для второго положения и ориентации камеры (a), и для эго-движения (сдвиг и вращение) (b) приведено для различных разностей высоты ландшафта

Так трансляционный компонент потока содержит информацию, требуемую для определения положения и ориентации камеры, было бы интересно наблюдать эффект от увеличения величины этого компонента. Последнее моделирование, представленное в этой работе, демонстрирует точность положения и ориентации камеры, когда поступательный компонент эго-движения изменяет величину от 5 м до 95 м. Хотя это изменение не влияет существенно на точность эго-движения, уменьшение неопределенности для оценок положения и ориентации камеры значительно для большой величины сдвигов (см. Рис. 15). Как заключение из вышеупомянутого можно заключить, что промежуток времени между двумя положениями камеры должен быть настолько велик, насколько алгоритм расчета оптического потока может позволить.

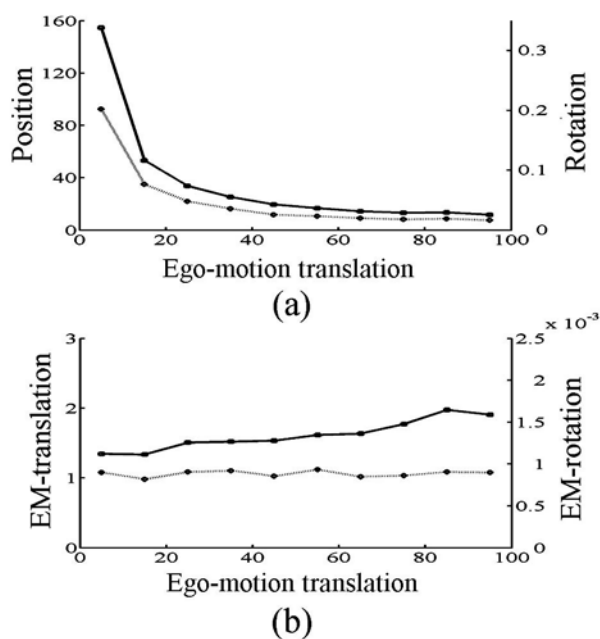


Рис. 15: Среднеквадратическое отклонение для второго положения и ориентации камеры (а), и для эго-движения (сдвиг и вращение) (б) приведено для различных величин поступательного компонента эго-движения

7.2 Результаты числового моделирования для реальных параметров полета и камеры.

Инерциальные навигационные системы (INS) обычно используются для обнаружения положения и ориентации ракет. Проблема этого метода состоит в том, что его ошибка увеличивается во времени. Мы предлагаем использовать новый метод (Навигационный Алгоритм, основанный на Оптическом потоке и Цифровой Карте Местности) [15], чтобы исправить результат INS и сделать ошибки конечными и постоянными. Фильтр Калмана используется, чтобы объединить результаты INS и нового метода [12]. Анализ погрешностей с линейной аппроксимацией первого порядка используется, чтобы найти корреляционную матрицу ошибок для нашего нового метода [14]. Мы сделали числовые моделирования полета с реальными параметрами полета и камеры, используя только INS, а затем INS и наш новый метод вместе, чтобы проверить полноценность этого нового метода.

Выбранные параметры полета следующие:

Высота полета 700, 1000, 3000 м.

Скорость полета - 200m/s.

Время полета - 800 s.

Траекторию полета мы можем видеть на (Рис. 16). Цифровая Карта Местности использовалась как ячейка (Рис. 17) для наших моделирований. Эта ячейка была продолжена периодически, чтобы получить полное ландшафт (Рис. 18). Случайный шум использовался как главный компонент шума INS. Более реальный шум дрейфа сдвига нуля дает намного большую ошибку (приблизительно 6000 м. вместо полученных 1000 м. в точке конца полета).

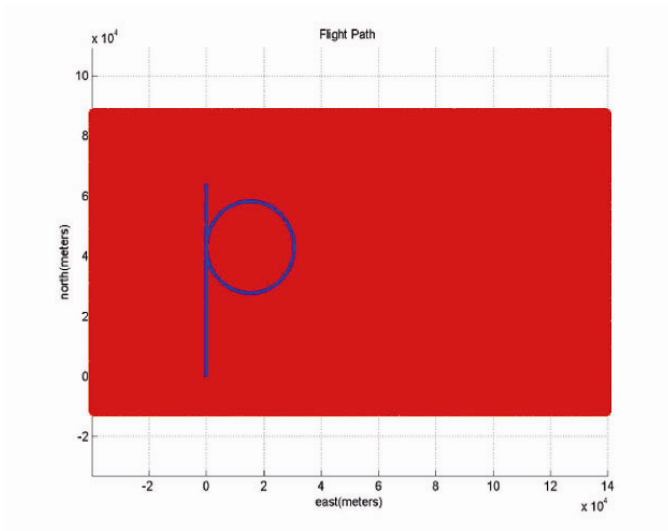


Рис. 16. Траектория полета.

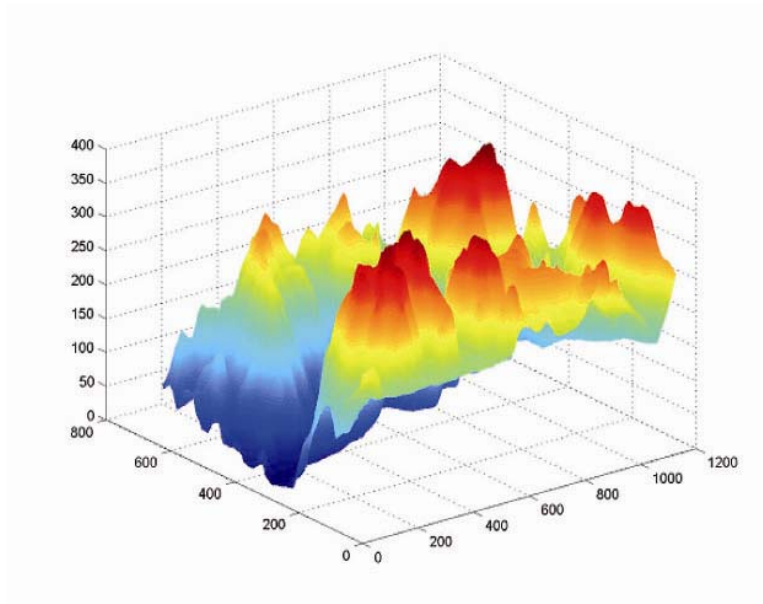


Рис. 17: Отображение действительного основания использовалось как ячейка.

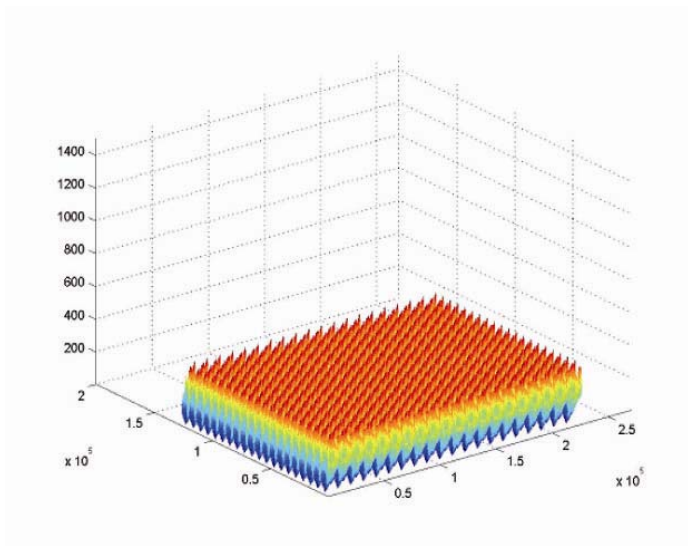


Рис. 18: Ячейка была продолжена периодически, чтобы получить полный ландшафт.

Выбранная камера и параметры моделирования следующие:

FOV (Угол зрения камеры) является 60° .

Число пар характерных точек, найденный на фотографиях, 100, 120.

Разрешение камеры 500x500, 1000x1000, 4000x4000. (Разрешение камеры определяет точность обнаружения характерных точек, мы не учитываем грубые ошибки для характерных точек).

расстояние между камерами (baseline) составляет 30 м., 50 м. или 200 м. (расстояние между камерами - расстояние между двумя положениями камеры, используемыми, чтобы сделать две фотографии для нового метода).

$\Delta time$ - 5s, 15 s, 30-ые. (временной интервал $\Delta time$ между измерениями).

Типичные результаты числовых моделирований приведены на (Рис. 3, 4, 5, 6) для различных случаев полета, параметров камеры и параметров моделирования. Ниже приведены таблицы ошибок для типичного случая с положительными результатами: x, y, z ошибки положения INS как с использованием нового метода, так и не используя новый метод.

Используемый полет, параметры камера и параметры моделирования для этого случая:

FOV - 60°

Число пар характерных точек, найденный на фотографиях 120

Разрешение 1000x1000

baseline 200 м.

$\Delta time$ 15 s.

Скорость полета составляет 200 м/с

Высоты составляют 700 м., 1000 м., 3000 м.

Таблица 1. ось X: Максимальная ошибка для INS с и без нового метода для различных высот.

Высота	700 м.	1000 м.	3000 м.
Максимальная ошибка x без нового метода	900 м.	130 м.	1300 м.
Максимальная ошибка x с новым методом	25 м.	20 м.	100 м.

Таблица 2. ось Y: Максимальная ошибка для INS с и без нового метода для различных высот.

Высота	700 м.	1000 м.	3000 м.
Максимальная ошибка y без нового метода	1000 м.	2000 м.	400 м.
Максимальная	25 м.	20 м.	100 м.

ошибка у с новым методом			
--------------------------	--	--	--

Таблица 3. ось Z : Максимальная ошибка для INS с и без нового метода для различных высот.

Высота	700 м.	1000 м.	3000 м.
Максимальная ошибка z без нового метода	250 м.	180 м.	250 м.
Максимальная ошибка z с новым методом	25 м.	20 м.	150 м.

Позвольте нам продемонстрировать ошибочные таблицы для типичного случая с положительными результатами: x, y, z ошибки положения INS с использованием нового метода для различных разрешений камеры. Используемый полет, камера и параметры моделирования для этого случая:

FOV 60 градусов, Номер features:120, Разрешение 500x500, 1000x1000, 4000x4000, Данные 200 м., Deltatime 15 s, скорость Полета 200 м\с, Высоты: 1000 м.

Таблица 4. ось X : Максимальная ошибка для INS без и с новым методом для различных разрешений камеры.

Разрешение	500x 500	1000x 1000	4000x4000
Максимальная ошибка x без нового метода	500 м	1000 м	4000 м
Максимальная ошибка x с новым методом	50 м.	20 м.	10 м.

Таблица 5. ось Y : Максимальная ошибка для INS без и с новым методом для различных разрешений камеры.

Разрешение	500x 500	1000x 1000	4000x4000
Максимальная ошибка y без нового метода	500 м	1000 м	4000 м
Макс y ошибка с новым методом	50 м.	20 м.	10 м.

Таблица 6. ось Z : Максимальная ошибка для INS без и с новым методом для различных разрешений камеры.

Разрешение	500x 500	1000x 1000	4000x4000
Максимальная	500 м	1000 м	4000 м

ошибка z без нового метода			
Макс z ошибка с новым методом	35 м.	20 м.	10 м.

8 Открытые вопросы и будущее развитие метода.

1) Если местоположение близко к вырожденному случаю (например, для небольшого угла зрения камеры, почти плоского ландшафта, небольшое число характерных точек и так далее), мы не можем использовать описанный метод, потому что невозможно найти сположения камер от этих данных. Но для этого случая возможно с помощью найденных соответствующих характерных точек, уточнить результаты INS помощью фильтр Калмана. Мы можем рассмотреть непосредственно эти соответствующие характерные точки (не вычисляя положение и ориентацию на основе этих характерных точек) как измерением для фильтра Калмана. Пример такого уточнения может быть найден в [16]. Но в этом случае ошибки метода увеличатся со временем, подобно INS. Так, после некоторого времени, положение слишком далеко от истинного положения, и мы не можем использовать ЦКЛ и имеем только эпполярные ограничения. Для описанного в статье метода ошибка остается постоянной, таким образом, мы способны использовать ЦКЛ все время.

2) Возможно рассмотреть более оптимальные и быстрые методы для того, чтобы искать минимум функции, дающей положение и ориентацию камеры. Например возможно улучшить начальное состояние для описанного метода, используя эпполярные уравнения (25) для вычисления R_{12} и p_{12} с точность до постоянной. Следующий шаг может быть использования уравнения (21) для вычисления R_1 . И заключительный шаг, использовать уравнение (18) для вычисления p_{12} и p_1 . Результат может быть улучшен описанным итеративным методом.

3) Мы можем искать не только некоторые случайные характерные точки. Так, вершины холмов, днища долин, а также холмы, закрывающие горизонт, могут использоваться для определения положения и ориентации.

4) использование пространственных (распределенных), а не точечных особенностей, а также распознавание некоторых характерных объектов.

5) Метод поиска не локального, а глобального минимума функции оценки в большой области

6) Поиск заданного участка местности в базе данных ЦКЛ обширных территорий, когда мы совсем не знаем наше местоположение

7) Использование описанных методов в различных практических целях: ориентация в комнатах, на улицах города, в теле человека.

9. Заключение

Был представлен алгоритм для определения положения и ориентации камеры и оценки движения, использующий характерные точки в изображениях и ЦКЛ, использующий фильтра Калмана. ЦКЛ служит глобальным ориентиром, и ее данные использовались для того, чтобы получить абсолютное положение и ориентацию камеры. Численное моделирование достаточно точно, чтобы предотвратить накопление ошибки и предотвратить дрейф траектории.

Анализ погрешностей был выполнен для нового алгоритма, который использует как оптический поток, полученный из двух последовательных изображений, так и ЦКЛ. Положение, ориентация и параметры эго-движения камера могут быть оценены предложенным алгоритмом. Главные источники ошибок были идентифицированы при вычислении оптического потока - качество информации о ландшафте, структура наблюдаемого ландшафта и траектория камеры. Был разработан анализ погрешностей. Обширные числовые моделирования были выполнены, чтобы изучить влияние вышеупомянутых коэффициентов.

Проверенный согласно разумным и общим сценариям, алгоритм вел себя сильно даже когда сталкивается с относительно шумной и стимулирующей окружающей средой. После анализа приходим к заключению, что предложенный алгоритм может эффективно использоваться как часть навигационной системы автономных транспортных средств.

На результатах числового моделирования для дреальных параметров полета и камеры мы также можем заключить, следующее:

1) Самый важный параметр моделирования – угол зрения камеры (FOV): для небольшого FOV метод расходится. Для FOV 60° результаты очень хороши. Причина для этого - то, что для небольшого FOV (12° или 6°) ситуация близка к вырожденной: мы должны выбрать небольшое смещение между камерами, наблюдаемый участок основания является слишком небольшим и почти плоским.

2) Разрешение камеры - также очень важный параметр: для лучшего разрешения у нас есть намного больше лучших результатов, из-за намного большей точности обнаружения особенностей.

3) Точность нового метода зависит от высоты полета. Первоначально точность увеличивается с увеличением высоты, потому что мы можем использовать большие данные и можем видеть больший участок ландшафта. Но на больших высотах точность начинает уменьшаться из-за эффекта параллакса.

Благодарность

Мы бы хотели поблагодарить Ронена Лернера, Эхуда Ривлина и Хектора Ротштейна за очень полезные консультации.

Библиография

[1] John Oliensis, "A critique of structure-from-motion algorithms", Computer Vision and Image Understanding, vol. 80, pp. 172–214, 2000

[2] Y. Liu, M.A. Rodrigues, "Statistical image analysis for pose estimation without point correspondences", Pattern Recognition Letters, vol. 22, pp. 1191-1206, 2001

[3] P. David, D. DeMenthon, R. Duraiswami, H. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination", ECCV 2002, LNCS 2352, pp. 698-714, 2002

[4] J. L. Barron and R. Eagleon, "Recursive estimation of time-varying motion and structure Parameters", Pattern Recognition vol. 29, no. 5, pp. 797--818, 1996

[5] T.Y. Tian, C. Tomashi, D.J. Hegger, "Comparison of approaches to egomotion computation", Department of Psychology and Computer science, Stanford university, CA 94305, 1996

[6] A.Chiuso, S.Soatto, "MFm: 3-D Motion From 2-D Motion, Causally integrated over time", Washington University Technical Report, 1999

[7] M. Irani, B. Rousso, S. Peleg, "Robust Recovery of Ego-Motion", Proc. Of CAIP 93, pp. 371-378, 1993

[8] D.G. Sim, R.H. Park, R.C. Kim, S.U. Lee, I.C. Kim, "Integrated position estimation using aerial image sequences", IEEE transactions on pattern analysis and machine intelligence, vol. 24, no.1, 2002

- [9] Technical report (to be added after the double-blind review).
- [10] R. M. Haralick, "Propagating Covariance In Computer Vision", *Advances in Image Understanding*, Azriel Rosenfeld, Bowyer and Ahuja, (eds.), IEEE Computer Society Press, pp. 142-157, Washington, 1996
- [11] Rees W.G., "The accuracy of Digital Elevation Models interpolated to higher resolutions", *International Journal of Remote Sensing* vol. 21 no.1, pp. 7-20, 2000
- [12] Greg Welch and Gary Bishop "An Introduction to the Kalman Filter" UNC-Chapel Hill TR 95-041, NC 27599-3175 April 5, 2004
www.menem.com/ilya/digital_library/control/welch-bishop-01.pdf
- [13] www.GPSSoftNav.com
- [14] O. Kupervasser, R. Lerner, E. Rivlin and P. H. Rotstein "Error Analysis for a Navigation Algorithm based on Optical-Flow and a Digital Terrain Map" *Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium*, P.1203-1212
- [15] R. Lerner, E. Rivlin and P. H. Rotstein "Pose and Motion Recovery from Feature Correspondences and a Digital Terrain Map" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1404-1417, Sept., 2006.
- [16] Pini Gurfil, Hector Rotstein "Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach", *Journal of Guidance, Control and Dynamics* 0731-5090 vol.24 no.5 (1016-1028), 2001
- [17] C. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 22(6):610–622, 2000.
- [18] D. Hoaglin, F. Mosteller, and J. Tukey, editors. *Understanding robust and exploratory data analysis*. John Wiley & Sons Inc, 1983.