# Error Analysis for a Navigation Algorithm based on Optical-Flow and a Digital Terrain Map

R. Lerner, E.Rivlin

Dept. of Computer Science
The Technion
Haifa 32000, Israel

P. H. Rotstein [*]

Dept. of Aerospace Engineering and Mechanics
University of Minnesota
Minneapolis, MN 55455

## 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 navigation 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. An example for such algorithm is [1]. 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 [2]. 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 [3], 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 [3] 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 [4].

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 un-

[*] on Sabbatical leave from Rafael,Israel

[1] Paper ID: 730

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)

certainty 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. Also supplied is the optical-flow field: $\{u_i(t_k)\}$ (*i=1...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:

$$N^T({}^{W}G - {}^{W}G_E) \approx 0 \qquad (1)$$

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

$${}^{W}G = R(t_1) \cdot {}^{C}q(t_1) \cdot \lambda \; + \; p(t_1) \qquad (2)$$

Here, $\lambda$ 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):

$$N^T(\lambda \cdot R(t_1) \cdot {}^{C}q(t_1) \; + \; p(t_1) \; - \; {}^{W}G_E) = 0 \qquad (3)$$

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

$$\lambda = \frac{N^{T\,W}G_E - N^T p(t_1)}{N^T R(t_1){}^{C}q(t_1)} \qquad (4)$$

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

$${}^{W}G = R(t_1){}^{C}q(t_1) \cdot \left( \frac{N^{T\,W}G_E - N^T p(t_1)}{N^T R(t_1){}^{C}q(t_1)} \right) + p(t_1) \qquad (5)$$

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

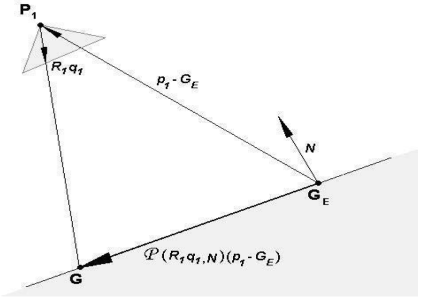

Figure 1: Geometrical description of expression (9) using the projection operator (7)

$\Delta p(t_1, t_2)$ 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 were 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:

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

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

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

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:

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

Using the projection operator, (8) becomes:

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

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:

$$^{C_2}G = p_{12} + R_{12} \left( R_1^T \left( G - p_1 \right) \right) \tag{10}$$

Assigning (9) into (10) gives:

$$^{C_2}G = p_{12} + R_{12} \mathcal{L} \left( G_E - p_1 \right) \tag{11}$$

$\mathcal{L}$ in the above expression represents:

$$\mathcal{L} = \frac{q_1 N^T}{N^T R_1 q_1} \tag{12}$$

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 $q_2$:

$$^{C_2}G = \frac{q_2}{|q_2|} \cdot \left( \frac{q_2^T}{|q_2|} \cdot {}^{C_2}G \right) \tag{13}$$

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:

$$\left[ \mathrm{I} - \frac{q_2 \cdot q_2^T}{q_2^T \cdot q_2} \right] \cdot {}^{C_2}G \quad = \quad \mathcal{P}(q_2, q_2) \cdot {}^{C_2}G \quad = \quad 0 \tag{14}$$

$^{C_2}G$ 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:

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

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. 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. The applicability, accuracy and robustness of the algorithm was verified though simulations and lab-experiments.

## 4. 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 (15):

$$\begin{aligned} f_i(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) &= \\ = \quad \mathcal{P}(q_2, q_2)\left[p_{12} + R_{12}\mathcal{L}_i\left(G_{E_i} - p_1\right)\right] & \qquad (16) \end{aligned}$$

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: $F(p_1, \phi_1, \theta_1, \psi_1, p_{12}, \phi_{12}, \theta_{12}, \psi_{12}) = [f_1, \ldots, f_n]^T$. 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 $\theta$ represents the 12-vector of the parameters to be estimated, and $D$ is the concatenation of all the data 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 [5] 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:

$$\Sigma_\theta = \left(\frac{dg}{d\theta}\right)^{-1}\left(\frac{dg}{dD}\right)\Sigma_D\left(\frac{dg}{dD}\right)^T\left(\frac{dg}{d\theta}\right)^{-1} \qquad (17)$$

Here, $\Sigma_\theta$ and $\Sigma_D$ represent the covariance matrices of the parameters and the data respectively. $g$ is defined as follows:

$$g(\theta, D) \doteq \frac{d}{d\theta}M(\theta, D) = \frac{d}{d\theta}F^T F = 2J_\theta^T F \qquad (18)$$

$J_\theta = dF/d\theta$ 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:

$$\frac{dg}{d\theta} \approx 2J_\theta^T J_\theta \qquad (19)$$

$$\frac{dg}{dD} \approx 2J_\theta^T J_D \qquad (20)$$

$J_D = dF/dD$ is defined in a similar way as the $(3n \times m)$ Jacobian matrix of $F$ with respect to the $m$ data components. Assigning (19) and (20) back into (17) yield the following expression:

$$\Sigma_\theta = \left(J_\theta^T J_\theta\right)^{-1} J_\theta^T \cdot \left(J_D \Sigma_D J_D^T\right) \cdot J_\theta \left(J_\theta^T J_\theta\right)^{-1} \qquad (21)$$

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

## 4.1. $J_\theta$ Calculation

Simple derivations of $f_i$ which is presented in (16), yield the following results:

$$\frac{df}{dp_1} = -\mathcal{P}(q_2, q_2)R_{12}\mathcal{L} \qquad (22)$$

$$\frac{df}{d\alpha_1} = -\mathcal{P}(q_2, q_2)R_{12}\mathcal{L}\left(\frac{d}{d\alpha_1}R_1\right)\mathcal{L}\left(G_E - p_1\right) \qquad (23)$$

$$\frac{df}{dp_{12}} = \mathcal{P}(q_2, q_2) \qquad (24)$$

$$\frac{df}{d\alpha_{12}} = \mathcal{P}(q_2, q_2)\left(\frac{d}{d\alpha_{12}}R_{12}\right)\mathcal{L}\left(G_E - p_1\right) \qquad (25)$$

In expressions (23) and (25): $\alpha_1 = \phi_1, \theta_1, \psi_1$ and: $\alpha_{12} = \phi_{12}, \theta_{12}, \psi_{12}$. The Jacobian $J_\theta$ is obtained by simple concatenation of the above derivations.

## 4.2. $J_D$ 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 form 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 $3 \times 3$ matrix $df_i/dq_{2_i}$ and $df_i/dG_{E_i}$for $J_q$ and $J_G$ respectively:

$$\frac{df}{dq_2} = \frac{-1}{\|q_2\|^2}\left[\left(q_2^T \cdot {}^{c_2}G\right)\mathrm{I} + q_2 \cdot {}^{c_2}G^T\right]\mathcal{P}(q_2, q_2) \qquad (26)$$

$$\frac{df}{dG_E} = \mathcal{P}(q_2, q_2)R_{12}\mathcal{L} \qquad (27)$$

${}^{c_2}G$ in expression (26) is the ground feature $G$ under the second camera frame as defined in (11).

### 4.3. $\Sigma_D$ 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 $\Sigma_D$ 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:

$$\Sigma_D = \begin{bmatrix} \Sigma_q & 0 \\ 0 & \Sigma_G \end{bmatrix} \tag{28}$$

In this work the errors of image locations and DTM height are assumed to be additive zero-mean Gaussian distributed with standard-deviation of $\sigma_I$ and $\sigma_h$ 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:

$$\Sigma_{q_i} = \sigma_I^2 \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \tag{29}$$

and $\Sigma_q$ is the matrix with the $\Sigma_{q_i}$'s along its diagonal.

In [6] 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 $\Sigma_G$ containing the $3 \times 3$ covariance matrices $\Sigma_{G_i}$ 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 $\lambda$, but due to the DTM height error the point $\tilde{G}_E = \left(\tilde{x}, \tilde{y}, \tilde{h}\right)^T$ 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:

$$N^T (G_E - H) = N^T (p_1 + \lambda R_1 q_1 - H) = 0 \tag{30}$$

Extracting $\lambda$ from (30) and assigning it back to $G_E$'s expression yields:

$$G_E = p_1 + R_1 \mathcal{L} (H - p_1) \tag{31}$$

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

$$\frac{dG_E}{dh} = R_1 \mathcal{L} \cdot \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T = \frac{R_1 q_1}{N^T R_1 q_1} \tag{32}$$

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:

$$\Sigma_{G_i} = \left(\frac{dG_E}{dh}\right) \cdot \sigma_h^2 \cdot \left(\frac{dG_E}{dh}\right)^T = \sigma_h^2 \cdot \frac{R_1 q_1 q_1^T R_1^T}{\left(N^T R_1 q_1\right)^2} \tag{33}$$

### 4.4. $\Sigma_{C_2}$ 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:

$$p_2 = p_1 - R_1 R_{12}^T p_{12} \tag{34}$$

$$R_2 = R_1 R_{12}^T \tag{35}$$

The uncertainty of the second pose estimates will be described by a $6 \times 6$ covariance matrix that can be derived from the already obtained $12 \times 12$ covariance matrix $\Sigma_\theta$ by multiplication from both sides with the second camera Jacobian $J_{C_2}$.

## 5. Simulations Results

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 $400 \times 400$ image resolution flies at altitude of 500m above the terrain. The terrain model dimensions are $3 \times 3$ km with 300m elevation differences (Fig.6(b)). A DTM of 30m grid spacing is being used to model the terrain (Fig.4(c)). The DTM resolution

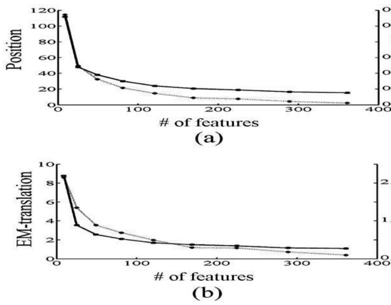

Figure 2: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the number of flow-features. In both graphs, the left vertical axis measures the translational deviations (in meters) and corresponds to the solid graph-line, while the right vertical axis measures the rotational deviations (in radians) and corresponds to the dotted graph-line

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.2, 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.2(a) presents the standard-deviations of the second frame of the camera while the deviations of the ego-motion are shown in Fig.2(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.3). 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.

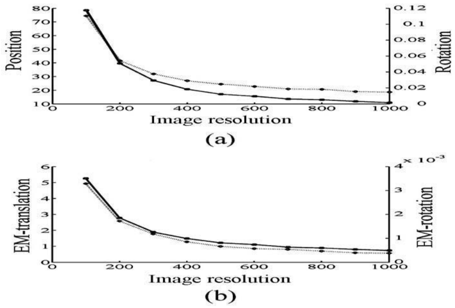

Figure 3: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the image resolution

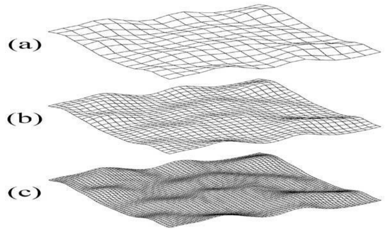

Figure 4: Different DTM resolutions: (a) grid spacing = 190m, (b) grid spacing = 100m, (c) grid spacing = 30m

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.4). 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.5). This phenomenon can be explained by the linear effect of the DTM resolution on its heights deviation.

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

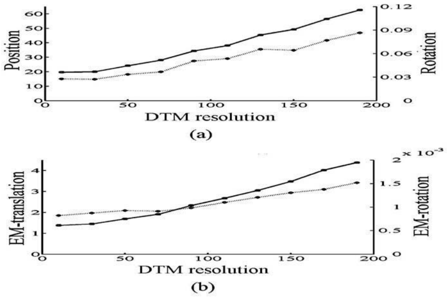

Figure 5: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the grid-spacing of the DTM

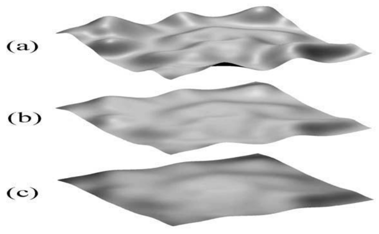

Figure 6: DTM elevation differences: (a) 150m, (b) 300m, (c) 450m

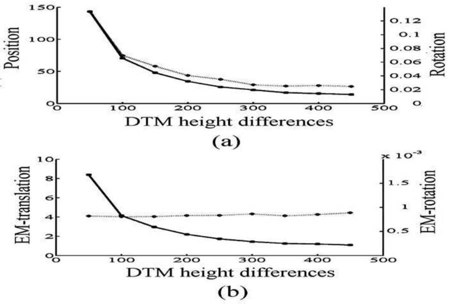

Figure 7: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the height differences of the terrain

ground features do not contain the required information for the camera pose 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 (fig. 7). For this simulation, the DTM elevation differences were scaled to vary from 50m to 450m (Fig.6). Note that the terrain structure has no direct effect on the rotational component of the ego-motion since this component, as was well established, is independent from the features' depths.

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.8). 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.

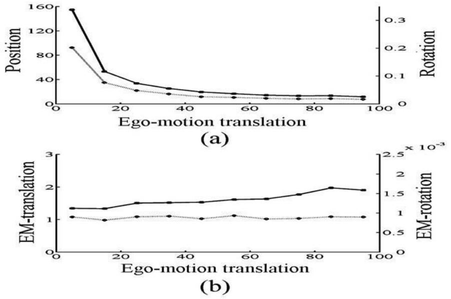

Figure 8: Average standard-deviation of the second position and orientation (a), and the ego-motion's translation and rotation (b) with respect to the magnitude of the translational component of the ego-motion

## 6. Conclusions

An error analysis has been performed for a novel algorithm that uses the optical flow and a DTM. The position, orientation and ego-motion parameters of the the camera can be estimated by the proposed algorithm. A closed form expression for the uncertainty of the pose and motion was developed. Extensive numerical simulations were performed to study the influence of different 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.

## References

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

[2] T. S. Huang, A. N. Netravali, "Motion and structure from feature correspondences: a review", Proc. Of the IEEE, vol. 82, no. 2, 1994

[3] 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

[4] John Oliensis, A critique of structure-from-motion algorithms, Computer Vision and Image Understanding, vol. 80, pp. 172214, 2000

[5] 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

[6] 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