# POSE ESTIMATION USING FEATURE CORRESPONDENCES AND DTM

*R.Lerner, E.Rivlin*

The Technion
Dept. of Computer Science
Haifa 32000, Israel

*P.H.Rotstein* *

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

## ABSTRACT

A novel algorithm for pose estimation using feature correspondences and Digital Terrain Map (DTM) is proposed. A constraint is formulated by using corresponding features from two consecutive frames together with the information provided by the elevation map. The proposed constraint is nonlinear and is solved by using a simple numerical method. The proposed approach does not require an intermediate explicit reconstruction of the 3D world. The feasibility of the algorithm is studied using both synthetic data of a virtual terrain and experimental data obtained from a terrain model and a robotic camera.

## 1. INTRODUCTION

This paper discuss a novel approach for estimating the position and orientation of a moving platform equipped with a camera using feature correspondences, and data obtained from 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.

There are two common approaches for the vision-based navigation problem: *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 [1], [2]. 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 corresponding features, 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 [3], [4]. The weakness of ego-motion integration comes from the fact that small errors from the ego-motion estimates 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 [5], such navigation-system is being suggested. In that work, like in this work, the drift is being corrected using a DTM. In [5] a patch from the ground was first reconstructed using structure-from-motion (SFM) algorithm and then matched to the DTM in order to derive the camera's pose. The patch reconstruction phase, which doesn't use the additional knowledge from the DTM, positions their technique under the same critique that applies for SFM algorithms [6].

In this work a novel algorithm for pose estimation and drift correction is described. The pose of the camera at two consecutive frames is derived using a DTM and the corresponding features in the two frames. The proposed algorithm does not require an intermediate explicit reconstruction of the 3D world. The algorithm is based on the following observation. Since the DTM supplies information about the structure of the observed terrain, each hypothesized pose of the camera will dictate the depths of the visible features. Hence, given the pose at two frames, the features' displacement can be uniquely determined. The objective of the algorithm will be finding the pose at the two frames which lead to features' displacement as close as possible to the given corresponding pairs. The algorithm enjoys the advantages and avoids the disadvantages of the two previously mentioned approaches. Although a drifted initial-guess is being used, no integration of previous results is made and the pose is being estimated directly. Unlike the landmarks approach no specific features should be detected and matched. Only the correspondence between the two consecutive images should be found.

*on Sabbatical leave from Rafael,Israel

0-7803-8554-3/04/$20.00 ©2004 IEEE.

## 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 consecutive time instances $t_1$ and $t_2$: At each of the two time instances a rough estimate of the camera's pose- $p_E(t_1)$, $R_E(t_1)$, $p_E(t_2)$ and $R_E(t_2)$ - is supplied (the subscript letter "$E$" denotes that this is an estimated quantity). Also supplied is the feature correspondences: $\{u_i(t_k)\}$($i$=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.

The objective of the proposed algorithm is to estimate the true camera's pose at the two frames: $p(t_1)$, $R(t_1)$, $p(t_2)$ and $R(t_2)$, using the feature correspondences, the DTM and the initial-guess.

## 3. DEFINING A CONSTRAINT FOR THE CAMERA'S POSE

In the following section, the corresponding image locations of a single feature will be used to define a constraint for the camera's pose at the two time instances mentioned above.

Consider an observed feature on the ground ${}^{w}G \in \mathbb{R}^3$. This feature is being projected on the camera's image-plane to the points $u(t_1)$ and $u(t_2)$ at the two time instances. Let ${}^{c}q(t_1)$ and ${}^{c}q(t_2)$ be the homogeneous representation of these locations: ${}^{c}q(t_k) = (u(t_k)\ ,\ 1)^T$. One can think of these vectors as being the vectors from the camera's optical-center to the projection point on the image plane. Using the initial-guess of the camera's pose at $t_1$ the line passing through $p_E(t_1)$ and ${}^{c}q(t_1)$ can be intersected with the DTM. Ray-tracing style algorithm can be used for this purpose. The location of this intersection will be marked as ${}^{w}G_E$. This ground-point is different from the true ground-feature ${}^{w}G$ (which was projected to ${}^{c}q(t_1)$) since an erroneous pose has been used in order to obtain it. However, for a reasonable initial-guess, ${}^{w}G_E$ and ${}^{w}G$ should be close enough. Hence, the DTM can be linearized around ${}^{w}G_E$. This leads to:

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

In this expression, $N$ denotes the normal of the DTM linearization plane. The true ground feature ${}^{w}G$ can be described using the first frame's true pose parameters:

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

$\lambda$ denotes the depth of the ground feature. Using (1) and (2):

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

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

(4) can be assigned back into (2) to obtain:

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

To simplify notations $X(t_k)$ will be marked as $X_k$ ($X = R$, $p$, $q$ ; $k$ =1, 2). The superscript describing the coordinates frame in which the vector is given will be dropped, except for cases were special attention needs to be drawn to the frames. Normally, $q$'s are given 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:

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

In order to further simplify the expressions a new projection operator is introduces:

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

This operator projects vectors on the subspace normal to $s$ along the direction of $u$. In order to demonstrate the above mentioned property one can 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 reordering one obtains:

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

Using the projection operator (8) becomes:

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

See Fig.1 for geometrical interpretation of the above expression.

The next step is transferring $G$ from the global coordinates frame- $W$ into the second camera's frame $C_2$. Since $p_2$ and $R_2$ describe the transformation from $C_2$ into $W$ the inverse transformation is used:

$${}^{c_2}G = R_2^T\,({}^{w}G - p_2) \qquad (10)$$

$q_2$ is the projection of the true ground-feature $G$. Thus, the vectors $q_2$ and ${}^{c_2}G$ should coincide:

$$\mathcal{P}(q_2, q_2) \cdot {}^{c_2}G \quad = \quad 0 \qquad (11)$$

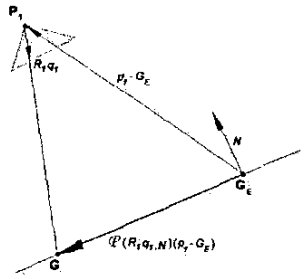


**Fig. 1**. Geometrical description of expression (9) using the projection operator (7)

Plugging (9) into (10) and then (10) into (11) the final constraint is obtained:

$$\mathcal{P}(q_2, q_2) \cdot R_2^T \left( \mathcal{P}(R_1 q_1, N)\,(p_1 - G_E) - (p_2 - G_E) \right) = 0 \tag{12}$$

This constraint involves the two positions and orientations 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 feature until a non-singular system is obtained. Twelve parameters need to be estimated (three for each position and orientation). Thus, at least six features are required for solving the system. Usually more vectors will be used in order to define an over-determined system which will lead to more robust solution. It is emphasized that the obtained constraint is non-linear. Thus, an iterative scheme will be used in order to solve this system.

## 4. ALGORITHM VALIDATION AND EXPERIMENTAL RESULTS

### 4.1. Algorithm's implementation

A robust algorithm which uses Newton-iterations and M-estimator was implemented to demonstrate the applicability and performance of the proposed approach. This standard optimization-scheme searches for the weighted-least-squared solution of the over-determined system, where the weights are being recalculated at each iteration by the M-estimator. Small weights are being assigned to large residuals which suppress the influence of outlier features.

Since the DTM linearization assumption is only approximately true this scheme can correct only part of the initial drift. In order to handle larger drifts the whole process should be repeated several times. Each time, the $G_E$'s are obtained by ray-tracing algorithm which uses the updated pose $C_1$. Next, the DTM is linearized around the new $G_E$'s, several Newton-iterations are executed, and the pose estimate is improved. From here on, the above procedure will be referred to as *external-iteration*, while the Newton-iterations will be referred to as *internal-iterations*. Since the ray-tracing is time consuming, it will be desirable to perform several internal-iterations before the next external one to improve as much as possible the pose's estimate and thus reduce the number of ray-tracing activations.

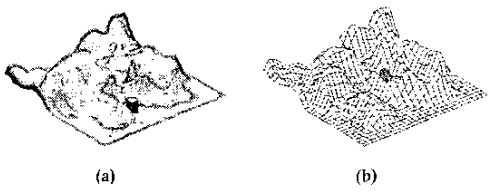


**Fig. 2**. (a) The virtual terrain, (b) The DTM constructed from this terrain. Notice the small bump in the two images places in different locations.

Two types of experiments were conducted: A simulation experiment with a synthetic data and lab-experiment using real data obtained by camera. The following two subsections discuss the settings for the two experiments and present the results that have been obtained.

### 4.2. Simulation results

For performing this experiment, a virtual-terrain of 300 × 300 meters containing hills as high as 60 meters was created. A high-resolution DTM was constructed using 1m grid spacing. In order to verify the robustness of the algorithm, the terrain was modified in order to defer from the DTM (see Fig.2). Virtual camera was placed in various positions and orientations and the locations of 100 corresponding features were analytically derived. The initial-guess was drifted by about 17m and 3 °. An example for the algorithm's convergence is shown in Fig.3. One can see that convergence is reached after four external-iterations. Although thirty internal-iterations were performed for each external-iteration, it is clear that few iterations would have been sufficient.

### 4.3. Lab experiment results

The second type of experiments was performed using a real 3D model of a terrain and real images obtained by a camera. The model's dimensions were 500 × 770mm with 240mm elevation differences (Fig.4(a)). By using a laser-based 3D-scanner, the shape of the terrain model was captured and a DTM was constructed using 10mm grid spacing (Fig.4(b)).

During the experiments, images of 1024 × 768 were obtained by a Dragonfly video camera at a rate of 15 frames per second. The camera was attached to a robotic arm which

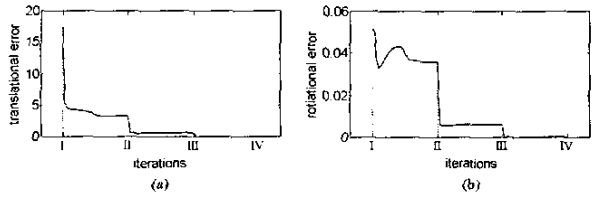


**Fig. 3**. Translational (a) and rotational (b) errors of the calculated pose with respect to the algorithm's iterations. I, II, III, and IV are external-iterations. Each iteration contains 30 internal-iterations. Measures are in meters and radians


**Fig. 4**. (a) The terrain's 3D model. Size = 500 × 770mm. (b) The DTM constructed by using of a laser-based 3D-scanner. *Grid spacing = 10mm*

provided the camera's pose at any moment. About 500 corresponding features were derived using Lucas-Kanade tracking method ([7]). Time gap of 2/15 seconds differentiated the frames couples.

In each experiment the camera was moved along a trajectory. An artificial drift was added to the path and the algorithm was operated at several points along the path to reduce the drift. The true, drifted and calculated trajectories for one of the experiments are shown in Fig.5. Similar results were obtained for the other trajectories. One can see that the algorithm achieves accurate estimates. In addition, the quality of the estimates does not decrease along time since the camera's pose was derived directly.

## 5. CONCLUSIONS

In this paper, a novel algorithm for pose estimation using feature correspondences and a DTM was presented. An intermediate explicit reconstruction of the 3D world was not required. A constraint was defined for each tracked feature, and the camera's pose was derived for the two frames involved. The robustness and accuracy of the algorithm were tested and verified using simulations and experiments. As was demonstrated throughout experiments with real data, the proposed algorithm may be used by navigation-systems of mobile platforms for eliminating the accumulated drift and reducing the errors significantly.

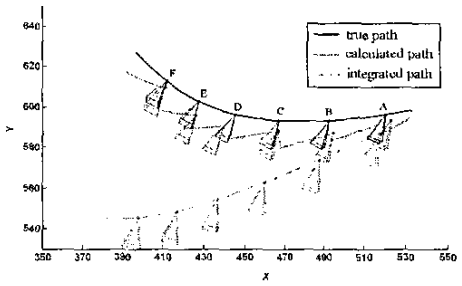


**Fig. 5**. The true, drifted and calculated trajectories of the camera taken from one of the lab experiments. At the six examined points, the translational-drift without using the algorithm was A:10.2, B:20.2, C:31.6, D:43.7, E:56.7, F:70.7, and has been reduced to A:4.9, B:7.0, C:6.2, D:11.0, E:11.7, F:3.6. The rotational-drift without using the algorithm was A:0.076, B:0.136, C:0.196, D:0.256, E:0.316, F:0.377, and has been reduced to A:0.008, B:0.019, C:0.008, D:0.036, E:0.035, F:0.008 (all measures are in millimeters and radians).

## 6. REFERENCES

[1] R. M. Haralick, C-N. Lee, K. Ottenberg, M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem", International Journal of Computer Vision, vol. 13, no. 3, pp. 331-356, 1994

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

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

[4] 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

[5] 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

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

[7] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", Im. Und. Wk., pp.121-130, 1981