The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Fast Pose Estimation For Visual Navigation Using Homographies

E. Montijano and C. Sagues
DIIS - I3A, University of Zaragoza, Spain
Email: {emonti, csagues} @unizar.es

Abstract— **In this paper we propose a new algorithm for relative pose estimation between two images based on a new decomposition for an homography matrix faster than the classical solutions. We introduce in our method approximate information about the planes in the reference images but this additional information allows the decomposition to avoid the multiplicity of solutions. An exhaustive analysis of the error propagation through the method is provided. The new decomposition can be used for visual navigation of robots moving on a planar surface. The approach is based in the well known teaching-by-doing scheme with the route defined by a set of images. The parameters of the planes can be computed automatically from the reference images before the navigation. The contributions are the easy method to extract the parameters of all the planes and the use of this information in navigation tasks using a fast homography decomposition. The experimental results show the performance of the proposal.**

Index Terms - homography, motion estimation, error analysis.

I. INTRODUCTION

Even though the autonomous navigation of mobile robots has been extensively studied it remains an open problem. Many repetitive tasks constantly made by humans moving from one place to another can be done by autonomous robots. Vision sensors provide a lot of information that can be used for this purpose. The appearance of invariant descriptors like SIFT [12] or SURF [1] in computer vision has made possible to achieve these tasks in a more precise and robust way.

The problem of navigation through a path defined by a set of images has also been explored by many researchers. Some of them have proposed qualitative solutions to make it. For example, in [14] a sequence of images combined with movement tags is used to repeat the route. In [4] another qualitative method in which the robot corrects its orientation using a voting system on the features is presented. In these solutions a problem appears when the robot gets lost due to obstacles or occlusions, because the robot will not be able to follow the path and it will not reach its goal.

Pose estimation can solve this problem allowing the robot to move through other routes by knowing at every moment its localization within the environment. SLAM methods based on Kalman filters [5] or Particle filters [7] estimate the robot pose while they map the environment providing good results using features. Such methods based on features and filters require a prior model of the robot motion and the images must be very close in order to perform a correct data association. Moreover, map maintenance requires additional computational time but when the robot is repeating a previously learnt path such a map maintenance is less important than reaction.

Geometric constraints like the epipolar constraint, defined by the Fundamental Matrix [9], can be imposed to the features to improve the estimations. In robot navigation Beardsley et al. have used it in [2]. However, when the parallax between images is small or the scene is composed by planes, the computation of the fundamental matrix becomes an ill-conditioned problem and it is no longer possible to apply this solution due to its instability. Nister has proposed an efficient solution using a five point algorithm [15] which avoids the planar singularity but the algorithm finds up to ten different possible solutions for two images, being required a third image to choose the right one. The five points algorithm has been successfully used for visual odometry with conventional and omnidirectional cameras [16], [10] and for robot localization in [17]. These approaches require bundle adjustment, which increases the computational time, or 3D information for triangulation, which increases the amount of information required.

Homographies are well suited for the situations where the epipolar constraint fails. An inter-image homography matrix, $\mathbf{H}$, defines a mapping relation between two images through points that belong to the same planar surface. It is well known that the estimation processes are improved in terms of accuracy and stability when considering the scene represented by planar regions [20]. A general decomposition of the homography to obtain the motion between two cameras can be found in [13]. Faugeras and Lustman describe in [8] a more specific decomposition for the homography when the camera is moving on a planar surface. Both approaches provide two possible solutions to the camera pose. Using these decompositions there have been different works related with visual navigation [6] or motion control [11].

In this paper we propose a technique based on homographies which intend to be useful in situations where the approaches commented previously are not well suited. The proposed decomposition is faster because it takes advantage of additional information available from the sequence of reference images, computed before the navigation. Preprocessing the reference sequence the method automatically obtains the parameters that define all the planes in the path giving only the scale factor, defined as the distance of the first plane to the first reference image. This information is used during the navigation so that the method always returns

This work was supported by the projects DPI2006-07928, DPI2009-08126, IST-1-045062-URUS-STP.

a unique solution. We also analyze the error propagation to the outputs from the error in the homography. The error transfer for our pose estimation algorithm is easy and also very efficient, which turns out to be another improvement respect to other approaches.

The Fast Homography Decomposition (Fast-HD) is presented in section II. How to automatically compute the normals and distances in the different reference frames is shown in section III. The error analysis of the method can be found in section IV. Experimental results are in section V. Finally in section VI we give the conclusions of this work.

II. Fast Pose Estimation using the Homography

The notation followed in this paper is the following: matrices are denoted as capital bold letters, $\mathbf{A}$, and vectors are named with non capital bold letters, $\mathbf{b}$. $\mathbf{A}_{(i)}$ represents the i-th row of $\mathbf{A}$, $\mathbf{A}_{(i,j)}$ represents element (i,j) of $\mathbf{A}$, and $\mathbf{b}_{(i)}$ the i-th element of $\mathbf{b}$. For estimations of the variables we use the hat, $\hat{\cdot}$. The errors in the estimations are represented using the tilde, $\tilde{\cdot} = |\cdot - \hat{\cdot}|$. We compute the Jacobian matrices of the vectors deriving by rows. Since the paper deals with several images and planes, for an easy understanding, subscript indices will correspond to images whereas superscript indices will correspond to planes in all the paper.

Let us suppose we have a set of reference images $\mathcal{I}_r \in \mathcal{I}$, captured by a robot in a teaching phase with the same camera, which has known calibration $\mathbf{K}$,

$$\mathbf{K} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

being $f_u > 0$ and $f_v > 0$ the focal length of the camera in pixel dimensions in the u and v direction respectively and (u_0, v_0, 1) the coordinates of the principal point. In a second stage the robot is moving autonomously acquiring real images. Without loss of generality $\mathcal{I}_1$ will represent the reference image whereas $\mathcal{I}_2$ will be the image acquired by the robot during the navigation. The motion between the two images, $\mathbf{R}_{21}$ and $\mathbf{t}_{21}$ is the unknown we want to solve. We assume the robot is moving on the XZ ground plane, which is reasonable for man-made environments. In this case only three parameters are needed to describe the state $\mathbf{s} = (x, z, \theta)^T$, with x and z the coordinates of the robot in the plane and θ its orientation. If one plane, $\boldsymbol{\pi}$ is visible in the two images, it is possible to compute a projective mapping (inter-image homography), $\mathbf{H}_{21}$, that relates the points belonging to the plane, $\mathbf{p}_2 = \mathbf{H}_{21}\mathbf{p}_1$, in both images. This homography is defined up to a scale factor and has the form

$$\mathbf{H}_{21} = \mathbf{K}(\mathbf{R}_{21} - \frac{\mathbf{t}_{21}\mathbf{n}_1^T}{d_1})\mathbf{K}^{-1}, \tag{2}$$

with d_1 and $\mathbf{n}_1$ the distance and normal of the plane in the reference frame respectively. Assuming the plane is not parallel to the plane of motion, the homography is normalized dividing by $\mathbf{H}_{(2,2)}$ given that this parameter never vanishes due to the planar motion constraint, leaving 8 constraints that can be used to estimate the motion.

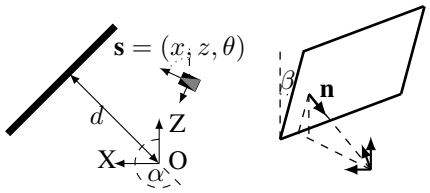


Fig. 1. Reference system

Any plane in the scene can be defined by its distance, d, and the unitary normal vector, $\mathbf{n} = (n_x, n_y, n_z)^T$, with respect to the reference frame. The planar motion assumption is used to reduce the parameters that define the planes and also to extract some properties of these parameters. The normal vector can now be defined with two angles, α and β, α depending on the reference orientation, and β a constant for each plane depending on its inclination. The normal is then expressed as $\mathbf{n} = (\sin\alpha, \cos\alpha\sin\beta, \cos\alpha\cos\beta)^T$ (Fig. 1). In the following of this section we assume that these parameters are known. The automatic computation of the plane parameters from the reference sequence is explained in the section III.

Passing the calibration matrices to the left side in (2) and clearing the unknowns, a linear system $\boldsymbol{\Pi}\mathbf{s}^* = \mathbf{b}$ is obtained with

$$\boldsymbol{\Pi} = \begin{pmatrix} -n_x & 0 & 0 & d_1 \\ -n_z & 0 & -d_1 & 0 \\ 0 & -n_x & d_1 & 0 \\ 0 & -n_z & 0 & d_1 \end{pmatrix}, \mathbf{s}^* = \begin{pmatrix} x_{21} \\ z_{21} \\ \sin\theta_{21} \\ \cos\theta_{21} \end{pmatrix} \tag{3}$$

and

$$\mathbf{b} = d_1 \begin{pmatrix} \mathbf{H}_{(1,1)} - \mathbf{H}_{(3,1)}u_0 \\ f_u^{-1}(u_0\mathbf{H}_{(1,1)} - \mathbf{H}_{(3,1)}u_0^2 + \mathbf{H}_{(1,2)}v_0 + \mathbf{H}_{(1,3)} - u_0\mathbf{H}_{(3,3)}) \\ \mathbf{H}_{(3,1)}f_u \\ \mathbf{H}_{(3,1)}u_0 + \mathbf{H}_{(3,3)} \end{pmatrix}, \tag{4}$$

$n_x = \sin\alpha$ and $n_z = \cos\alpha\cos\beta$ and the additional constraints in the homography matrix

$$\begin{aligned} &\mathbf{H}_{(1,2)} = -x_{21}n_y f_u/f_v d_1, \quad &&\mathbf{H}_{(3,2)} = 0, \\ &\mathbf{H}_{(2,1)} = \mathbf{H}_{(3,1)}v_0, &&\mathbf{H}_{(2,3)} = \mathbf{H}_{(3,3)}v_0 - v_0 \end{aligned} \tag{5}$$

Let us notice that $\boldsymbol{\Pi}$ is only plane dependent and not singular because its determinant is equal to $d_1(n_x^2 + n_z^2)$, being zero when $\alpha = k\pi$ and $\beta = (2k+1)\pi/2, k \in \mathbb{N}$, which by hypothesis does not happen. This means that we can analytically compute its inverse and solve the linear system in constant time, $\mathbf{s}^* = \boldsymbol{\Pi}^{-1}\mathbf{b}$. The orientation is obtained as $\theta_{21} = \arctan 2([\boldsymbol{\Pi}^{-1}\mathbf{b}]_{(3)}, [\boldsymbol{\Pi}^{-1}\mathbf{b}]_{(4)})$. We consider both unknowns $\sin\theta_{21}$ and $\cos\theta_{21}$ instead of only one because it clears the ambiguity of dual solutions dealing with angles. The dual solution for $\mathbf{R}$, $\mathbf{n}$ and $\mathbf{t}$ does not appear using this method since we are introducing the information of $\mathbf{n}$ in the equations. The additional constraint automatically solves this ambiguity and we also do not need to compute any eigenvalues as other classical homography decompositions usually do [21], [8] but to solve a linear system of rank 4 which clearly improves the previous approaches. The pose robot expressed in the reference coordinates can be obtained by an inverse transformation, $\mathbf{s}_{12} = \mathbf{s}_{21}^{-1}$

Since the method assumes the calibration to be known, one natural question that arises is why not computing the homography with the unprojected points, $\mathbf{p}_u = \mathbf{K}^{-1}\mathbf{p}$, which will make the vector $\mathbf{b}$ and the pose estimation equation easier. We have discarded this approach for two reasons. The first one is the computation time. The cost of multiplying all the features in one image (assuming the reference features have been previously unprojected) by the inverse of the calibration matrix is more expensive than the direct computation of the homography using the image coordinates. The second reason is the precision in the homography. The possible errors that may exists in the calibration parameters will not affect to the homography and will only affect to the pose estimation whereas if we compute the homography with the real coordinates the possible errors in the calibration would also affect the homography making the pose estimation less reliable.

III. Processing the reference sequence

As we have seen, in order to use the Fast-HD, it is necessary to know the plane parameters in the reference images, $\mathbf{n}$ and d. These parameters can be estimated in an offline stage before the autonomous navigation. Let us consider as input data the sequence of reference images and the distance of the first plane to the first reference image. The distance must be manually introduced to solve the scale factor. It is well known that from monocular images it is not possible to recover the scale factor. With this information the normal of the reference plane in the first reference can be estimated. From there, the parameters of the same plane or any others that may appear in the sequence can be easily estimated. In this section the method to compute the plane parameters is presented.

A. Obtaining the normal in the first reference

In order to estimate the first normal the method computes an homography between the two first initial images. In this first case a classical homography decomposition is performed obtaining the two possible solutions for $\mathbf{R}_{21}$, $\mathbf{n}^1$ and $\mathbf{t}_{21}$. The constraint that all the points must have positive depths, $(\mathbf{n}^1)^T\mathbf{K}^{-1}\mathbf{p} > 0$, is imposed in order to discard the illusive (wrong) solution from the correct one. Once the real solution has been computed the angles that define the plane in the reference image, α_1^1 and β^1, are extracted from the normal vector. For the next reference images the process can be done using the Fast-HD, obtaining directly the correct solution.

B. Updating the parameters for the next references

With the parameters of one plane in one reference it is possible to compute the parameters of the same plane in other reference images. Using the data from the first reference image the position of the second reference is computed using the Fast-HD. Once the relative position between the two images is known the parameters of the plane in the second image are computed as (Fig. 2)

$$\begin{cases} \alpha_2^1 = \alpha_1^1 - \theta_{12} \\ d_2^1 = d_1^1 + z_{12}\cos\alpha_1^1 + x_{12}\sin\alpha_1^1. \end{cases} \quad (6)$$

The process is repeated for all the reference images where the plane is visible. Let us note that β^1 remains constant for every position so that update its value is not necessary.

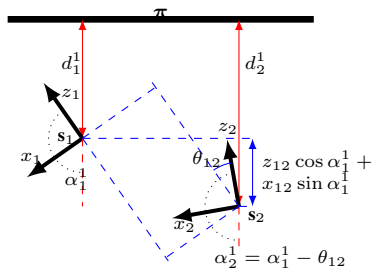


Fig. 2. Estimation of the plane parameters in a second reference from the parameters in the first one.

C. Computing the parameters for other planes

In normal situations there will be several planes visible in the path. In these situations multi-plane constraints are used to compute the parameters of a second plane knowing the parameters of the first one. Assuming that every plane is visible with at least another plane in two or more reference images it is possible to compute two homographies, $\mathbf{H}_{21}^1$ and $\mathbf{H}_{21}^2$, one for each plane. By using the two homographies an homology matrix is computed as $\boldsymbol{H} = (\mathbf{H}_{21}^1)^{-1}\mathbf{H}_{21}^2$. The homology captures the relative motion between the images through two planes visible in the two images. The homology is decomposed using the Sherman-Morrison formula [18] as $\boldsymbol{H} = \mathbf{I} + \mathbf{v}\mathbf{p}^T$ where

$$\begin{aligned} \mathbf{v} &= (v_1, v_2, v_3)^T = \mathbf{K}\frac{\mathbf{R}_{21}^{-1}\mathbf{t}_{21}}{1+\frac{(\mathbf{n}_1^1)^T}{d_1^1}\mathbf{R}_{21}^{-1}\mathbf{t}_{21}}, \\ \mathbf{p} &= (p_1, p_2, p_3)^T = (\frac{(\mathbf{n}_1^1)^T}{d_1^1} - \frac{(\mathbf{n}_1^2)^T}{d_1^2})\mathbf{K}^{-1}. \end{aligned} \quad (7)$$

The motion between the two images, $\mathbf{R}_{21}$ and $\mathbf{t}_{21}$, is computed using the parameters of the known plane. Then, using eq. (7) the normal of the second plane is cleared (normalized since d_1^2 is not known yet)

$$\mathbf{n}_1^2 = \mathbf{v}^{-T}(\mathbf{H} - \mathbf{I})\mathbf{K}^{-1} - \mathbf{n}_1^1, \quad (8)$$

being $\mathbf{v}^{-T} = (1/v_1, 1/v_2, 1/v_3)$. With the second normal computed, the second distance is obtained through the translation between the two images obtained with the second homography.

With these two expressions all the parameters of all the planes visible in all the reference images are computed and autonomous visual navigation of a robot using the Fast-HD is possible.

IV. Error propagation

Vision sensors, as any other sensor, are not 100% accurate so the measurements obtained will introduce error in the computation of the camera locations. It is interesting to analyze the sensitivity of the pose estimation using the Fast-HD to the sensor errors. This knowledge can be very useful during navigation, for example if the sensor needs to be

integrated with an Extended Kalman Filter (EKF), which needs an estimation of the measurement noise. We have analyzed how the pose estimation is affected by the errors in the homographies and the first-order approximation of the error is given.

A. Pose errors caused by homography errors

The errors in the homography could come from different causes like the discretization of the pixels. In order to estimate the errors in the homography, the errors in the feature coordinates during the acquisition of the images are considered. Let us assume that each pair of matched features $\mathbf{p} = (u, v, 1)^T$ and $\mathbf{p}' = (u', v', 1)^T$ has the perturbations $\tilde{\mathbf{p}} = (\tilde{u}, \tilde{v}, 0)^T$ and $\tilde{\mathbf{p}}' = (\tilde{u}', \tilde{v}', 0)^T$ respectively. From this assumption the error in the parameters of the homography using the DLT algorithm [9] is derived. Let us recall that the homography is computed as the eigenvector, $\mathbf{h}$, associated with the least singular value of the matrix $\mathbf{A}$, which depends on the point matches. The perturbation in the point coordinates creates a perturbed matrix $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{\Delta_A}$, where, assuming there are n matches, $\mathbf{\Delta_A}$ is a 2nx9 matrix. For each match the two corresponding rows of $\mathbf{\Delta_A}$ are

$$\mathbf{\Delta_{A}}_i = \begin{pmatrix} \mathbf{0}^T & \tilde{\mathbf{p}}_i^T & \tilde{v}_i'\mathbf{p}_i^T + v_i'\tilde{\mathbf{p}}_i^T \\ \tilde{\mathbf{p}}_i^T & \mathbf{0}^T & \tilde{u}_i'\mathbf{p}_i^T + u_i'\tilde{\mathbf{p}}_i^T \end{pmatrix} \tag{9}$$

It is proved [21] that the linear error propagated through the eigenvalue decomposition is

$$\tilde{\mathbf{h}} = \mathbf{G}\mathbf{\Delta}\mathbf{G}^T\mathbf{\Delta}_{\mathbf{A}^T\mathbf{A}}\mathbf{h}, \tag{10}$$

with $\mathbf{G}$ being the matrix of eigenvectors of $\mathbf{A}^T\mathbf{A}$, $\mathbf{\Delta}_{\mathbf{A}^T\mathbf{A}} = \mathbf{\Delta}_{\mathbf{A}}^T\mathbf{A} + \mathbf{A}^T\mathbf{\Delta}_{\mathbf{A}}$, and $\mathbf{\Delta}$ being a function of the eigenvalues λ_i,

$$\mathbf{\Delta} = diag\{(\lambda_9 - \lambda_2)^{-1}, (\lambda_9 - \lambda_3)^{-1}, \ldots, (\lambda_9 - \lambda_8)^{-1}, 0\}. \tag{11}$$

More information about the errors in the homographies can be found in [3].

The possible errors in the observations, $\tilde{\mathbf{s}}$, caused by the errors in the homography can be approximated by the second term of the Taylor series expansion of $\mathbf{s}_{12}$ multiplied by the errors in the homography,

$$\tilde{\mathbf{s}}_{\mathbf{h}} = \frac{\partial \mathbf{s}_{12}}{\partial \mathbf{h}}\tilde{\mathbf{h}}. \tag{12}$$

We can express $\mathbf{s}_{12} = \mathbf{s}_{12}(\mathbf{s}^*)$ and the pose estimation system as $\mathbf{s}^* = \mathbf{s}^*(\mathbf{h})$. Applying the chain rule we have

$$\frac{\partial \mathbf{s}_{12}}{\partial \mathbf{h}} = \frac{\partial \mathbf{s}_{12}}{\partial \mathbf{s}^*}\frac{\partial \mathbf{s}^*}{\partial \mathbf{h}}. \tag{13}$$

The first partial derivative has the form

$$\frac{\partial \mathbf{s}_{12}}{\partial \mathbf{s}^*} = \begin{pmatrix} -\cos(\theta_{21}) & -\sin(\theta_{21}) & z_{12} \\ \sin(\theta_{21}) & -\cos(\theta_{21}) & -x_{12} \\ 0 & 0 & -1 \end{pmatrix}. \tag{14}$$

In order to obtain the second Jacobian, we have $\mathbf{s}^* = (s_1^*, s_2^*, s_3^*)^T$, with

$$\mathbf{s}_1^* = [\mathbf{\Pi}^{-1}\mathbf{b}]_{(1)}, \mathbf{s}_2^* = [\mathbf{\Pi}^{-1}\mathbf{b}]_{(2)}, \mathbf{s}_3^* = \arctan(\frac{[\mathbf{\Pi}^{-1}\mathbf{b}]_{(3)}}{[\mathbf{\Pi}^{-1}\mathbf{b}]_{(4)}}). \tag{15}$$

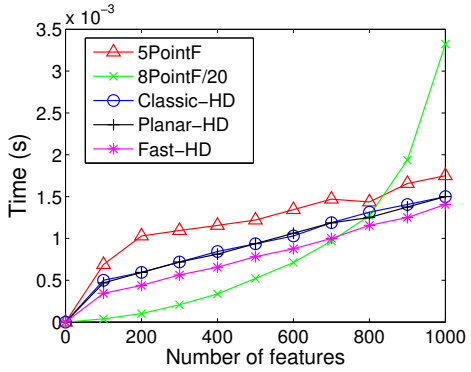


Fig. 3. Comparison of the time cost of several visual pose estimation techniques. 8-Point algorithm cost has been scaled dividing by 20 due to its high cost with respect to the other four algorithms. Fast-HD is faster than of all of them.

Since $\mathbf{\Pi}^{-1}$ does not depend on $\mathbf{h}$, the components of $\partial \mathbf{s}^*/\partial \mathbf{h}$ can be expressed as

$$\frac{\partial \mathbf{s}^*}{\partial \mathbf{h}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -[\mathbf{\Pi}^{-1}\mathbf{b}]_{(4)} & [\mathbf{\Pi}^{-1}\mathbf{b}]_{(3)} \end{pmatrix} \mathbf{\Pi}^{-1}\frac{\partial \mathbf{b}}{\partial \mathbf{h}}, \tag{16}$$

with

$$\frac{\partial \mathbf{b}}{\partial \mathbf{h}} = \begin{pmatrix} 1 & 0 & 0 & -u_0 & 0 \\ u_0/f_u & v_0/f_u & 1/f_u & -u_0^2/f_u & -u_0/f_u \\ 0 & 0 & 0 & f_u & 0 \\ 0 & 0 & 0 & u_0 & 1 \end{pmatrix}. \tag{17}$$

Multiplying all the matrices a 3x5 matrix is obtained, which multiplied by the homography error, $\tilde{\mathbf{h}}$, gives the error in the pose estimation, $\tilde{\mathbf{s}_{12}}$.

V. Experimental Results

We have performed a complete battery of tests, including both simulations and experiments with a real robot. Simulations have been used to analyze the behavior of the proposal and to compare its performance with previous approaches found in the literature while experiments in a real platform have been used to test its reliability.

The simulations have been carried out with synthetic data using Matlab in a Pentium with 2.82 GHz quad-core processor and 4Gb RAM. The first experiment has been designed to measure the computational times of different pose estimation algorithms depending on the number of features. We have compared two algorithms based on the fundamental matrix and three based on the homography matrix. The chosen algorithms are the eight point algorithm (8PointF) [9], the 5-point algorithm (5PointF) [15] (the code has been downloaded from `http://vis.uky.edu/~stewe/FIVEPOINT/` [19]), classical decomposition of the homography (Classic-HD) [13], homography decomposition when there is planar motion (Planar-HD) [8] and the Fast-Homography decomposition proposed (Fast-HD). For each estimation we have run 500 iterations computing the mean time. The algorithms receive all the matched points, where there are no outliers so we have not considered the time involved in RANSAC

Algorithm	Miss	Wrong **t**	Wrong θ	$\bar{\mu}_t$ (m)	$\bar{\mu}_\theta(rad)$	σ_t^2	σ_θ^2
8PointF	18.6%	73.7%	59.2%	0.2905	0.0270	0.2484	0.0210
5PointF	16.8%	20.4%	20.6%	0.1403	0.0157	0.1547	0.0175
Classic-HD	16.5%	4.9%	9.1%	0.0616	0.0061	0.0934	0.0093
Planar-HD	16.5%	4.7%	9.1%	0.0637	0.0064	0.0963	0.0100
Fast-HD	16.5%	4.3%	8.8%	0.0585	0.0055	0.0911	0.0101

TABLE I

ACCURACY OF DIFFERENT POSE ESTIMATION METHODS. FIRST COLUMN SHOWS THE PERCENTAGE OF TRIALS WHERE NO **H/F** WAS CALCULATED. WRONG **T** (WRONG θ) IS THE PERCENTAGE OF TRIALS WHERE THE TRANSLATION/ROTATION WAS ESTIMATED WITH AN ERROR HIGHER THAN 10% OF $|\mathbf{s}_1 - \mathbf{s}_2|$. $\bar{\mu}_t$ ($\bar{\mu}_\theta$) AND σ_t^2 (σ_θ^2) ARE THE MEAN AND STANDARD DEVIATION ERROR IN THE TRANSLATION/ROTATION ESTIMATION CONSIDERING ONLY THE TRIALS THAT DO NOT GIVE A WRONG ESTIMATION.

computations in this test. The three algorithms based on homographies use DLT algorithm to compute the homography. Respect to pose estimation, for all the algorithms we compute all the possible solutions and choose the one closer to the real solution except for our method, which automatically returns the right solution, since it is unique. Figure 3 shows the obtained results. The 8 point algorithm has a higher cost compared with the other four methods (up to 35 ms) so the time has been divided by twenty to fit it in the graphic. We can see that the Fast-HD decomposition is faster than the other methods.

The next experiment deals with the accuracy of the different pose estimation methods. Table I shows the accuracy of the chosen methods for 1000 trials with random positions in a 5x5 square and maximum rotation deviation of 0.65 radians observing a planar surface. The only error in the features is the one caused by the discretization of the pixels. We have measured the number of times each method cannot compute the position (miss), the times each one computes a position with error higher than 10% with respect to the distance between the reference image and the actual one (Wrong **t**), rotation with error higher than 10% (Wrong θ) and the mean errors and standard deviations of both translation and rotation when the pose estimation is under 10% error. We can see that homography algorithms have better performance than fundamental matrix algorithms because of short baseline between the images and because the features are in planes. The Fast-HD has lower mean errors and also computes a good position more often than the other algorithms.

We have also performed experiments using a real platform to test the reliability of the Fast-HD in real situations where there are noisy features and the images contain non planar regions. The experiments have been carried out with a Pioneer 3Dx inc. robot with a Canon VC-C4 PTZ camera previously calibrated that takes images of 640x480 pixels with a frame rate of about 1 image per second. SURF features [1] have been extracted and DLT+RANSAC [9] algorithm has been used to compute the homographies between the images. We present here two different scenarios with the different experiments performed. The low frame rate makes the approaches that use filters to provide bad pose estimations. The experiments present two situations in which our method is well suited but other approaches are not due to the presence of planar regions and the wide baseline between consecutive images.

In the first scenario (Fig. 4), we have moved the robot following several times an approximate square trajectory of 1.5 meters side using landmarks in order to maintain the path. We have roughly moved the robot using a joystick performing different kind of motions and slightly different routes. The scale factor has been coarsely estimated by counting the number of floor tiles from the plane to the position where the reference images were taken. Eight images have been used as references and the plane parameters were computed with a reasonable error using the method in section III. Next, we have compared the Fast-HD pose estimations with those given by the robot odometry (bottom graphic of Figure 4). The graphic plots the positions estimated by the Fast-HD (red line) and the odometry measures (blue line); it is observed that whereas the odometry goes through the walls and inside the square, the Fast-HD gives always positions in the trajectory or near it.

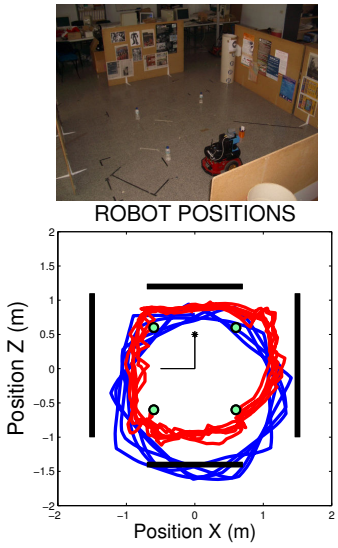
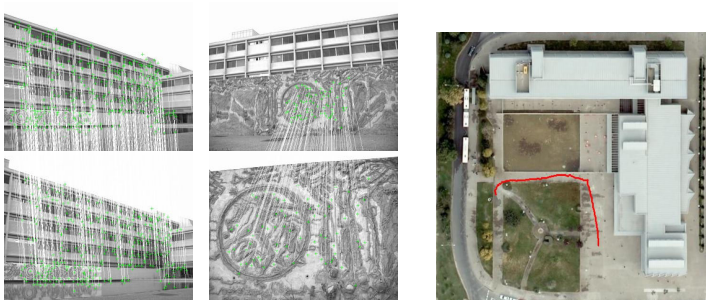


Fig. 4. Results of experiment in the laboratory. Using four reference images the robot performs 6 laps to the square trajectory. The upper photo shows a general view of the environment. The bottom figure shows the pose estimation given by the odometry (blue) and by our sensor (red).

The second experiment has been performed outdoors. We have moved the robot outdoors through the entrance of the Ada Byron building (where we work) describing a path of about 75 meters. In this case the reference planes are the



(a) (b)

Fig. 5. (a) Images of the trip around the Ada Byron building. Upper images are the actual images whereas bottom images are the images used as references. (b) Estimation of the route followed (red line).

walls of the building and we have not used any reference images. Instead of that we have given the robot the distance to the plane seen in the first image of the path and with the second image the normal has been estimated. After that the robot has been estimating its position with the Fast-HD using the calibrated images while trying to detect new planes when the homographies are not supported by enough matches between the images. The information about new planes has been estimated using (8). When the computation of one homography has not been possible from the first reference image, the robot picks the previous image acquired during the navigation and estimates its motion, using this image as a new reference for the following steps. Two images of the trajectory are shown in Fig. 5-a, the upper images are the current images acquired by the robot and the bottom images are the past images used as references. Figure 5-b shows the estimated trajectory using the Fast-HD overlapped with an image of the environment. The submitted video shows the results of this experiment. Due to size limitations the quality of the video is not very good. Another version with higher resolution can be downloaded from http://webdiis.unizar.es/~edumonti/?p=103.

VI. Conclusions

We have presented a Fast Homography Decomposition suited for mobile robot visual navigation in man-made environments. The method exploits the automatically previously computed information about the planes in the scene. We have compared its performance with other pose estimation methods found in the literature obtaining a similar accuracy but in less time. The method gives robot pose solving a linear system in constant time which is computationally more efficient and avoids the multiplicity of solutions. An exhaustive analysis of the robustness of the method to the possible errors in the homography has been done. The experimental results show the performance of our method in different environments and real situations where other approaches may fail due to planar scenes or wide baseline.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[2] P.A. Beardsley, A. Zisserman, and D.W. Murray. Navigation using affine structure from motion. In *European Conference on Computer Vision (ECCV)*, pages 85–96, 1995.

[3] P. Chen and D. Suter. Error analysis in homography estimation by first order approximation tools: A general technique. *Journal of Mathematical Imaging and Vision*, 33(3):281–295.

[4] Z. Chen and S.T. Birchfield. Qualitative vision-based mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1702–1708, May 2006.

[5] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Tra. on Robotics*, 55(5):372–382, 2008.

[6] J. Courbon, Y. Mezouar, and P. Martinet. Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25(3):253–266, 2008.

[7] E. Eade and T. Drummond. Scalable monocular slam. In *In Proc. Comp. Vision and Pattern Recog.*

[8] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[10] J.Tardif, Y.Pavlidis, and K.Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2531–2538, 2008.

[11] G. Lopez-Nicolas, C. Sagues, and J.J. Guerrero. Homography-based visual control of nonholonomic vehicles. *IEEE International Conference on Robotics and Automation*, pages 1703–1708, 2007.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[13] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[14] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *IEEE Int. Conf. Rob. and Autom.*, pages 83–88, 1996.

[15] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[16] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):756–770, 2006.

[17] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *Int. J. Comput. Vision*, 74(3):237–260, 2007.

[18] C. Sagüés, A.C. Murillo, F. Escudero, and J.J. Guerrero. From lines to epipoles through planes in two views. *Pattern Recognition*, 39(3):384–393, 2006.

[19] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, JUN 2006.

[20] R. Szeliski and P. H. S. Torr. Geometrically constrained structure from motion: Points on planes. In *In European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE*, pages 171–186, 1998.

[21] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin-Heidelberg, 1993.