Mapping Large Environments with an Omnivideo Camera

I. Esteban, O. Booij, Z. Zivkovic, B. Krose
IAS-TNO, ISLA, ISLA, IAS
iesteban,obooij,zivkovic, krose@science.uva.nl

***Abstract*— We study the problem of mapping a large indoor environment using an omnivideo camera. Local features from omnivideo images and epipolar geometry are used to compute the relative pose between pairs of images. These poses are then used in an Extended Information Filter using a trajectory based representation where only the robot poses corresponding to captured images are reconstructed. The features with the geometric constraints also give a robust similarity measure that is used for data association. Our experiments show that an accurate map can be built in real time of a small office environment. For large environments, big loops can be closed and a map can be built in nearly linear time.**

I. INTRODUCTION

SLAM stands for Simultaneous Localization and Mapping [1][2]. SLAM represents the process in which a robot drives around an unknown environment taking measurements with a certain sensor device. These measurements are then used to both estimating its own position within the environment while building a consistent map of it. Despite the fact that the basic SLAM framework was presented long ago [3], many new developments have been published that aim at the definition of a practical, simple and affordable solution. The SLAM framework can be described in terms of its challenges as: *measuring the environment, data association and computational costs*.

The trend during the last years with respect to the sensing device has been on using cameras as the principal measurement source [4][5][6][7][8][9]. Furthermore, the use of omnidirectional [10] [11] [12] cameras present some advantages with respect to monocular vision or stereo vision due to the large field of view. This simplifies the task of detecting when a previous are has been visited and has been used with success in the task of robustly detecting loop closures [13] and creating topological maps [13][11]. Although laser based SLAM seems to be the standard accurate solution, it is not suitable for very large environments as the number of features in the map grows rapidly.

Data association is the task of detecting when the robot is revisiting a previously seen area. This is necessary in order to detect when a loop has been closed and in order to bound the growth of the error in the estimation of the map. Data association is usually computationally expensive and methods exist that exploit both probabilistic information [14] [7] of the current location, and observations analysis [15] [9] [16] to obtain measurement matches.

Finally, the computational cost has received a lot of attention due to the limitations of the standard Kalman Filter (KF) solution. State augmentation, sparsification, particle filters or sub-mapping (see [2] for a complete review) are some of the most recent approaches that try to overcome these computational limits. Extended Information Filters for trajectory based representations [17] [14] [18] have received special attention due to their computational advantages and their potential to solve large mapping problems.

In this paper we present our Omnidirectional SLAM System for mapping large indoor environments. We use local image features, extracted from omnidirectional images, and epipolar geometry to obtain geometrical constraints between different robot poses. These, together with odometry measurements, are then used within an Extended Information Filter [17] to estimate both the robot's trajectory and the error in its estimation. As presented by Eustice et. al. [17] [14] [18] , EIF for trajectory based representations offer computational advantages, however, they require the state recovery process which is also computationally expensive. They propose a method for partial state recovery that introduces new errors in the estimated state. This method, though computationally interesting it is not suitable for loop closure as the information introduced at the loop closing points is disregarded. We explore the use of 5 different techniques to perform state recovery and analyze their performance for the task of large scale mapping. Further, we consider the problem of loop closure and solve the data association by estimating the similarity between current and previous images [13]. This is essentially different as in [17] as we do not use any pose information to search for image features matches. The complexity of a naive approach for data association where all images are compared to all other images is $O(n)$ while the complexity of EKF is $O(n^2)$, hence we do not consider the computational costs of data association and only focus on detecting loop closure situations.

This paper is organized as follows. Firstly we present our trajectory based map representation and how measurements are obtained from omnidirectional images. Secondly we describe the theoretical background of the Extended Information Filter and how both odometry and measurements are used to estimate the robot's trajectory. Thirdly we describe our experimental setup and detail our results for two different experiments. Finally we point out some conclusions and future work.

II. MAPPING LARGE ENVIRONMENTS

In this section we detail our definition of the map we are going to build using the trajectory based SLAM representation, also termed View based SLAM [17]. We first

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 297-306

define the state we want to estimate (the robot pose and the map) and then we describe how to obtain information from omnivideo images taken at every robot location to improve such estimate.

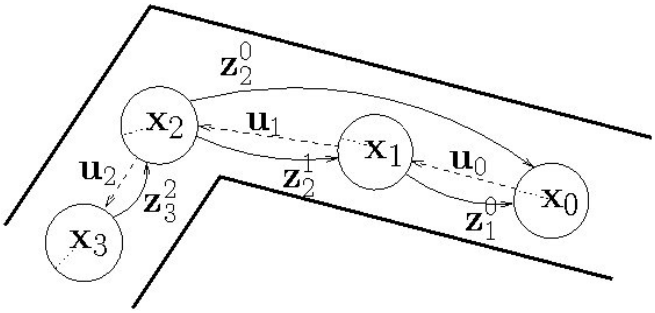

Fig. 1. Sample robot trajectory with odometry u and observations z

A. States

In conventional landmark based SLAM the state $\mathbf{x}$ that needs to be estimated consists of only the most recent robot pose as well as a growing set of 2D or 3D positions of landmarks reconstructed from the raw sensor data. This set of landmarks is called *the map*. In Trajectory SLAM the landmarks are not explicitly modeled, rather, the state at time step t consists of current and all previous robot poses. In our case these are the 2D positions and orientation angles: $\mathbf{x}_t = [\mathbf{x}_t^*, M]^T = [\mathbf{x}_0, \mathbf{x}_1, \ldots]^T = [x_0, y_0, \theta_0, x_1, y_1, \theta_1, \ldots]^T$. The first pose in $\mathbf{x}_t$ is always the current robot pose, also called $\mathbf{x}_t^*$ while the remaining of the poses represent the map M. This trajectory based representation reduces the complexity of the SLAM method as described in Section III.

Figure 1 visualizes an example operation of the Trajectory SLAM with 4 robot poses. The first robot pose $\mathbf{x}_0$ is added to the state (process called state augmentation) as $[0,0,0]^T$ and thus defines the coordinate frame for the rest of the state. An omnidirectional image is taken at this position. The state is then augmented with a new robot pose $\mathbf{x}_1$ using the odometry readings $\mathbf{u}_0$, also called the control vector. Again, an image is taken which is then compared with the image from the previous pose providing additional positional information $\mathbf{z}_1^0$, the observation vector, that is used to improve the estimate of the state. In $\mathbf{x}_2$ this procedure is repeated, matching with both previous robot poses. Then the robot drives around the corner to $\mathbf{x}_3$, causing the overlap of the new image with the first two images to reduce. The image comparison method, as explained in II-B, detects this and the pose estimation of $\mathbf{x}_3$ will be based on the observation of the last pose $\mathbf{z}_3^2$ and the odometry reading $\mathbf{u}_3$. An important aspect of any SLAM method that is not in this small example is the problem of data association (detecting when an image has been seen before). We solve this by using a robust image similarity measure as explained below.

An alternative approach to any SLAM mapping technique is the so called topological mapping [11] [13]. T.Goedeme et al. use omnidirectional image to build a topological map of an indoor environment. Topological mapping is conceptually closer to the way humans build spatial representations but offer significant disadvantages in terms of real applications where metrics are required to the robots to operate. On the other hand, a topological map is very sensitive to the imaging sensor (lightning conditions, cluttered spaces, etc) while a filtering approach is less sensitive as it filters out the inadequate measurements. In fact, we also build a topological map as we define links between successive images given their similarity. The difference with respect to other topological mapping approaches is that we build a metric map on top of them.

B. Observations

An observation $\mathbf{z}$ describes the relative positional information extracted from the current omnidirectional image and all the previous images which depict the same part of the environment. It is well known that the relative pose can be estimated from two images using the epipolar constraint [19]. Even though work has been published in the definition of the epipolar geometry for omnidirectional cameras [10], we use an entirely different approach. In [10][11] the epipolar geometry is defined to the very specific case of a central catadioptric camera with an hyperbolic mirror. Given this definition, 8 point correspondences are used to estimate the essential matrix. We approach the problem using regular epipolar geometry instead. Given the known calibration parameters of the camera and the exact shape of the hyperbolic mirror, we re-project the image to a cylinder around the ray between the center of the mirror and the center of the camera (see figure 2). This cylindrical image can then be treated a a regular image and the essential matrix can be estimated with a regular approach. The reason to use this approach instead of the more specific version of Svoboda is the fact that we can then use a standard fast SIFT implementation, whereas using the central catadioptric epipolar geometry will require the definition of a different descriptor for the image features. Hence, the estimation of the relative pose between two images is performed in the standard way by first extracting a set of local image features (computation $O(1)$) from both omnidirectional images which are then compared to give a set of n 3d point correspondences, $\mathbf{p}_1, \ldots, \mathbf{p}_n$ on one image surface and $\mathbf{q}_1, \ldots, \mathbf{q}_n$ on the other (see figure 2). In our experiments we used SIFT features (Scale Invariant Feature Transform) [20]. Point correspondences that resulted from the same world point in the environment can be related by the essential matrix which describes the relative camera pose: $\mathbf{p}_i^T E \mathbf{q}_i = 0$ for all i. Based on this function and the assumption that the robot moved on a planar surface, the matrix E can be estimated from 3 correspondences using the planar constrained 8-point algorithm [19]. To be robust against false point correspondences we use the planar constrained 8-point algorithm inside the hypothesize and test method RANSAC (RANdom SAmple Consensus). This provides us with the matrix E and the number of correspondences for which the re-projection error given E is small. If the ratio between this number and the number of features found in the images

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 297-306

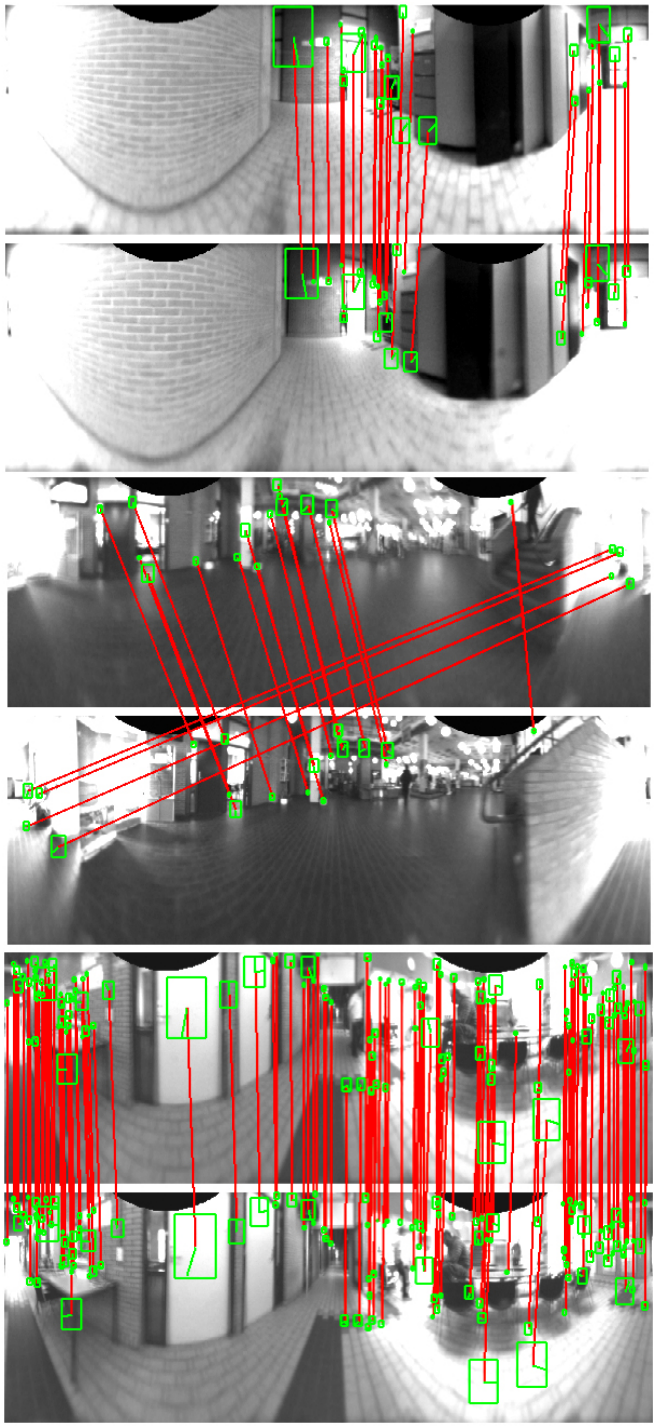

Fig. 2. Matching features in 3 typical office environments. TOP: typical narrow corridor. MIDDLE: entrance to a room after a corridor. BOTTOM: wide room.

is bigger than a certain threshold, which we set to 0.1 in our experiments, then we extract the pose information from E. Otherwise, we do not use this image pair for the observation. In this way we use the same algorithm for both determining which measurements to add to the observation vector, solving the data association problem, as computing the measurements themselves.

From E the relative pose can be extracted using [21] and results, in the case of 2D motion, in the direction of the translation ϕ and a 2D rotation θ. Both parameters are used to improve the estimate of the robot trajectory. The observation vector thus gets the following form: $\mathbf{z} = [\phi_0, \theta_0, \phi_1, \theta_1, \ldots]^T$.

This data association approach costs $O(n)$ in the number of images. A more interesting approach could be employed that takes advantage of a graph representation of the image similarity as shown in [22]. In that case, the computational costs of the data association step are lower than $O(n)$.

In [11] a two layer approach is used for image matching. First, a set of global features such as color histograms are used to determine a base similarity between two images. In this similarity measure is large enough, a set of local features is the extracted and compared to finally determine if the two images are sufficiently similar. As we use re-projected cylindrical images, we can use SIFT features and RANSAC to successfully discard false matches.

It is worth mentioning that our approach for relative pose estimation is valid as we obtained the calibration parameters from the manufacturer of the mirror and the camera and only the center of the mirror needed to be estimated to correctly re-project the images to a cylinder. If calibration parameters were not know, B.Micusik [12] proposes a simple yet accurate auto-calibration procedure where the circular field of view of the camera is used to obtain the calibration parameters, including the relative pose. Also, they use bundle adjustment to further improve the estimate. We believe that our results will be further improved by the use of such technique.

III. Exactly Sparse EIF

We have described the representation of our map x and the set of measurements z. We now use an Extended Information Filter to estimate both the map and the error covariance matrix.

The EIF is a mathematical equivalent to the well known Extended Kalman Filter. It is based on the information form or canonical representation of a Gaussian distribution, hence the state and error covariance are not directly estimated, but instead their *relatives* the information vector and information matrix.

$$\Lambda_t = \Sigma_t{}^{-1} \tag{1}$$

$$\eta_t = \Lambda_t \mu_t \tag{2}$$

The EIF works in a similar fashion as the EKF. First, using information from the odometry readings, the state

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 297-306

is estimated as described in Section II-A. Then, using the observations obtained through the images, this estimate is improved.

In this section we will first describe our motion and measurement models. Then the state augmenting procedure is detailed, and the computational implication of the use of the information vector and information form. We also show how the new measurements are introduced to improve the estimates. Finally, we introduce the problem of state recovery and propose five different practical solutions.

A. Motion and measurement models

The use of the EIF requires the definition of a motion model for the motion prediction step and a measurement model for the measurement update step. Most SLAM approaches require both models to be non-linear (hence the use of the Extended version of the filter to linearize those processes). However, as our robot did not have the raw odometry measurement readily available but only the variation in translation and heading ($u_t = [\Delta x, \Delta y, \Delta\alpha)$, we use a linear motion model:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + u_t + w, \tag{3}$$

where w is the odometry error model (white Gaussian noise with zero mean and covariance matrix Q).

For the measurement model and due to the nature of the observations (see Section II-B) we defined a non linear measurement process based on the transfer function $h(x)$. The function h transforms the state to measurements (measurement prediction):

$$\begin{aligned} \mathbf{z}_i &= h(\mathbf{x}_t, \mathbf{x}_{\mathbf{z}_i}) + v & (4)\\ &= \begin{bmatrix} atan2(y_{\mathbf{z}_i} - y_t, x_{\mathbf{z}_i} - x_t) - \theta_t \\ \theta_{\mathbf{z}_i} - \theta_t \end{bmatrix} + v, & (5) \end{aligned}$$

where $\mathbf{z}_i$ is the observed robot pose, $\mathbf{z}_i$ is the measurement prediction given the state $\mathbf{x}_t$ and v is the measurement error model (white Gaussian with zero mean and covariance matrix R).

As part of the EIF, the non-linear measurement process is linearized using a first order Taylor series:

$$\mathbf{z}_i \approx h(\hat{\mu}_t, \hat{\mu}_{z_i}) + H_i \begin{bmatrix} \mathbf{x}_t - \hat{\mu}_t \\ \mathbf{x}_{z_i} - \hat{\mu}_{z_t} \end{bmatrix}, \tag{6}$$

where H_i is the Jacobian of the function h evaluated at the mean.

B. Augmenting the state

In the EKF, the state is augmented with a new robot pose at every time step. This procedure can be simply described in terms of the new state vector and error covariance matrix:

$$\begin{aligned} \mu'_{t+1} &= \begin{bmatrix} \mu_{x_t^*} + \mathbf{u}_{t+1} \\ \mu_{x_t^*} \\ \mu_M \end{bmatrix} \\ \Sigma'_{t+1} &= \begin{bmatrix} (\Sigma_{x_t^* x_t^*} + Q) & \Sigma_{x_t^* x_t^*} & \Sigma_{x_t^* M} \\ \Sigma_{x_t^* x_t^*} & \Sigma_{x_t^* x_t^*} & \Sigma_{x_t^* M} \\ \Sigma_{M x_t^*} & \Sigma_{M x_t^*} & \Sigma_{MM} \end{bmatrix}, \end{aligned} \tag{7}$$

where Q is the error model for the odometry ($3x3$ diagonal matrix).

Using the alternative information form to describe the Gaussian distribution, we know that the estimate at time t is described by an information form Gaussian distribution as:

$$p(\mathbf{x}_t^*, M \mid \mathbf{z}_t, \mathbf{u}_t) = \mathcal{N}^{-1}\left(\begin{bmatrix} \eta_{x_t} \\ \eta_M \end{bmatrix}, \begin{bmatrix} \Lambda_{x_t x_t} & \Lambda_{x_t M} \\ \Lambda_{M x_t} & \Lambda_{MM} \end{bmatrix} \right)$$

Again, when a new robot pose is reached the robot state needs to be augmented:

$$p(\mathbf{x}_{t+1}^*, \mathbf{x}_t^*, M \mid \mathbf{z}_t, \mathbf{u}_{t+1}) = \mathcal{N}^{-1}\left(\eta'_{t+1}, \Lambda'_{t+1}\right) \tag{8}$$

Taking the previous standard representation of this distribution used in the EKF, and the formal relation between the normal form and the information form (see equations 1 and 2) we reach a state augmentation scheme by means of inverting the augmented error covariance Σ'_{t+1} (equation 7), obtaining the information vector η'_{t+1} and information matrix Λ'_{t+1}:

$$\begin{aligned} \eta'_{t+1} &= \begin{bmatrix} Q^{-1}(f(\mu_x, \mathbf{u}_{t+1}) - \mu_{x_t}) \\ \eta_{x_t} - Q^{-1}(f(\mu_x, u_{t+1}) - \mu_{x_t}) \\ \eta_M \end{bmatrix} \\ \Lambda'_{t+1} &= \begin{bmatrix} Q^{-1} & -Q^{-1} & 0 \\ -Q^{-1} & \Lambda_{x_t x_t} + Q^{-1} & \Lambda_{x_t M} \\ 0 & \Lambda_{M x_t} & \Lambda_{MM} \end{bmatrix} \end{aligned}$$

Note the zeros that result from augmenting the state in the information form. This is the key result that leads to the computational gain as the matrix is naturally sparse. When the state vector is augmented to include the new robot position $\mathbf{x}_{t+1}$ only shared information between the current robot pose and the previous robot pose is introduced. The shared information between the map M and the new robot pose is always zero. The only exception occurs when a loop is closed. In such situation, non diagonal elements will appear in the information matrix as shared information between the new robot pose and previously visited robot poses is introduced. If we would continue to introduce new states, we shall observe that the information matrix will present a block tridiagonal structure where each state is only linked to the previous and following one.

When we marginalize the state $\mathbf{x}_t^*$ from the distribution in equation 8 to perform the motion prediction, it can be proved [17] that it can be implemented in constant time as only a fixed portion of the information matrix is involved in the marginalization calculation.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice (Italy) 2008 November 3-4
ISBN 978-88-95872-01-8
pp. 297-306

Having seen the state augmentation in the time update step in both the covariance and information form, we can similarly obtain the expression in the information form for the measurement update step [17]:

$$\begin{aligned} \eta_t &= \hat{\eta}_t + H^T R^{-1}(\mathbf{z}_t - h(\hat{\mu}_t) + H\hat{\mu}_t) \\ \Lambda_t &= \hat{\Lambda}_t + H^T R^{-1} H \end{aligned}$$

This description of the measurement update step in the information form shows that the information matrix is additively updated by the product $H^T R^{-1} H$. As the jacobian H is always sparse and the amount of observations can be considered constant [17], only some elements of the information matrix need to be modified, hence the updates are *constant in time*.

Up to this point, the total complexity of the information filter equations is constant in time ($O(1)$) as opposed to the quadratic complexity of the standard EKF.

C. State Recovery

The drawback of the EIF is the fact that we no longer obtain an estimation of the state and its error covariance but we do obtain their *relatives* in the information form. Once the information matrix and information vector are obtained we need to transform them back to the original covariance and mean form in order to perform the following state augmentation.

The most naive recovery method inverting the information matrix results in cubic complexity and voids the efficiency obtained with the new information filter. In fact, recovering the state mean can be described as solving a sparse, symmetric, positive-definite, linear system:

$$\Lambda_t \mu_t = \eta_t, \tag{9}$$

In our Matlab experiments we tested five different state recovery techniques. All these techniques are generic solutions to solving the set of equations defined by $Ax = b$.

- **Inversion**: just to use as baseline for improvement measurement, we use the naive inversion technique which uses Gaussian elimination with partial pivoting. This is far from optimal and will make the experiments on large data-sets impractical.
- **CGS** (conjugate gradients squared method): this is an iterative solution to the set of equations. It requires the definition of an initial guess. In our case we used the state vector of the previous time step augmented with a new robot pose and a tolerance value of $1e-7$. The solution is not guaranteed to be exact.
- **PCG** (preconditioned conjugate gradients method): if the information matrix is ill conditioned, a preconditioner can be used to speed up the process. For our experiments we used the incomplete Cholesky factorization of the information matrix as preconditioner with a tolerance of $1e-7$. The solution in this case is also not guaranteed to be exact.
- **LU** (Lu decomposition): this method decomposes the information matrix into two triangular matrices, one of them with values in the upper right diagonal, and the other one on the lower left diagonal. It is an exact solution and works by solving the system for each of the triangular matrices.
- **Cholmod2** (supernodal sparse Cholesky backslash): this is not a method but rather an implementation of a solution using the Cholesky decomposition for sparse matrices. It is similar to the LU decomposition but in this case the two triangular matrices are the transpose of each other. It is part of the package SuiteSparse [1] by Tim Davis which is available on-line.

IV. Experiments and Results

Our approach to the SLAM problem did face three different challenges: computational complexity of the standard EKF solution, adequacy of an omnivideo camera to measure the environment and the loop closing problem. In order to test the performance of our omnivision trajectory based EIF approach, we carried out a set of experiments to demonstrate how our approach can solve each of the three problems.

Two sets of experiments are discussed. The first experiment illustrates the ability of our approach to build consistent and accurate maps of small environments. The second experiment was designed to measure the ability of our method to close large loops where the odometry error is very large and demonstrate the important computation gain of the EIF with respect to the standard EKF solution.

Doe to the difficulty to obtain ground truth data for our experiments, we only present laser data for the small scale experiment. This can be used to visually inspect the accuracy of our omnidirectional approach. For the large scale experiment, we are only concern with the computational costs of the state recovery and the loop closure scenario, hence no ground truth is presented.

A. Experimental Setup

For our experiments we used a Nomad Scout robot equipped with an omnivideo system consisting of a one mega-pixel firewire camera and an Accowle convex hyperbolic mirror. Additionally, and for visualization purposes only, the robot was equipped with a laser range scanner. The measurements taken with the laser were then used after the trajectory was corrected with our method to visualize the environment and illustrate the improvement in the accuracy of the map.

Due to the large field of view of the camera it was possible to generate 360 degrees images. Two images per second were recorded while the robot was driving at a maximum speed of 20 cm per second, resulting in an average of 1 image every 10 cm. A large portion of the long hallways where the robot was driven were poorly illuminated, posing a real challenge for the image matching algorithm. We measured the amount

[1] SuiteSparse is a collection of packages for working with exactly large sparse matrices. It is freely available on-line at: http://www.cise.ufl.edu/research/sparse/SuiteSparse/

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November 3-4
ISBN 978-88-95872-01-8
pp. 297-306

of light in some of these corridors obtaining an equivalent amount of light to a living room lit by Christmas tree lights (30 lux).

The odometry error covariance matrix Q was set to $[1mm^2, 1mm^2, 2deg^2]$ in the diagonal, and zeros in the non-diagonal. For the measurement error, we used a covariance matrix R with $[15deg^2, 15deg^2]$ in the diagonal and zeros in the non-diagonal.

We collected datasets in two different environments. A first dataset was collected by driving the robot manually in a small loop around two different offices and a small portion of the hallway that communicates them. The trajectory was driven twice in order to test our approach when previously seen areas are revisited. The total trajectory consist of 875 images taken at every robot pose along with odometry measures. Due to the smooth surface and the limited size of the environment, we artificially increased the odometry error by making the robot's wheel slip at some points of the trajectory to better illustrate the improvement in the accuracy of the SLAM corrected map.

The second data set was collected in the same office environment but driving a much larger loop. The robot was driven for more 1.5 hours along corridors and offices over a surface of more than $10,000$ squared meters. In order to test the ability of our approach to cope with closing large loops, we drove to the starting point after 45 minutes and having already taken $5,210$ images. The robot was then driven over the same part of the corridors to increase the overlap and finalized at the end of a corridor after having taken $10,325$ images in total. After the complete data-set was recorded, the accumulated error in the odometry added up to more than 80 meters in distance and 100 degrees in heading.

B. Small Office Environment

We present two maps of the same trajectory. Each map contains the same information, namely, the estimated trajectory of the robot shown as black circles, the laser data obtained at every robot location and the connected graph that represents the images that were found to have sufficient similarity shown as light gray lines connecting robot poses.

As we can see in figure 3, the images taken at sections of the trajectory where the robot drives multiple times over a hallway or office, are correctly matched. No false links are present in this data-set.

The first map (see figure 3) was built using odometry as the only information source. Despite the fact that some structure can be distinguished in the map (walls, doors and hallways), the odometry error adds up yielding duplicated structures (see figure 5, Left). If the robot would continue driving the same space for more loops, the accumulated error will make the map completely cluttered.

The second map (see figure 4) represents the same trajectory corrected with our omnivision trajectory based SLAM approach. Firstly, it is clear that our method can cope with loop closure in small environments as the duplication of structures in loop closing points is no longer present (see figure 5, Right) and previously seen areas are correctly

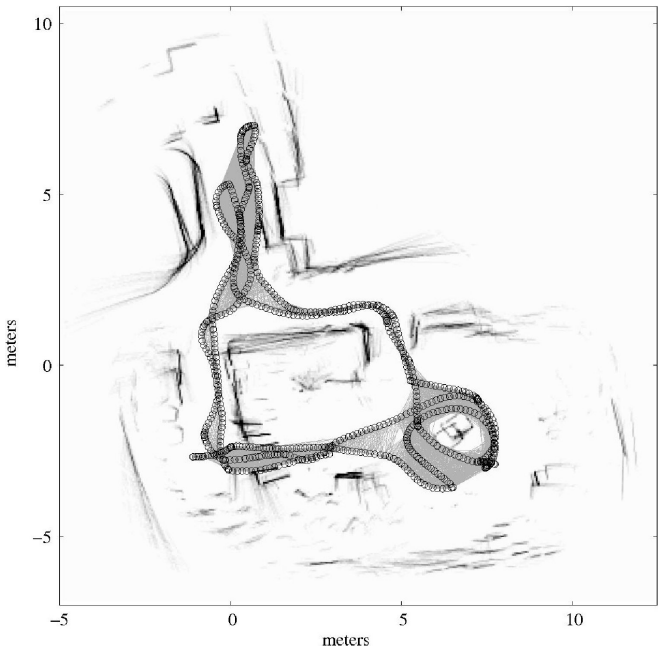

Fig. 3. Odometry based map with laser data and connectivity map for robot poses - small office

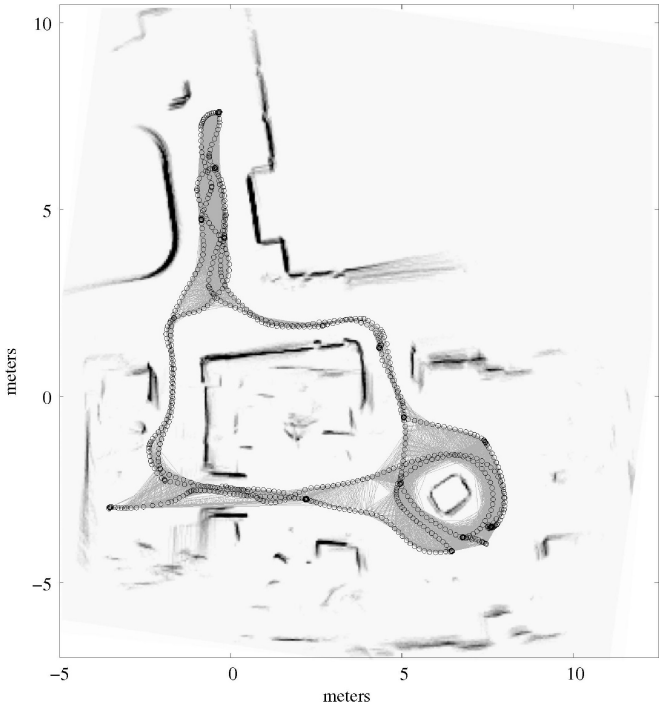

Fig. 4. Omnivision SLAM corrected map with laser data for visualization purposes - small office

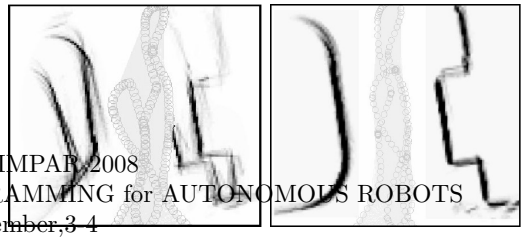

Fig. 5. **Left**: Odometry map, **Right**: SLAM map - small office

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 297-306

matched together. Secondly, the fact that no duplicated structures are present at all in the corrected map implies that the information regarding the relative pose obtained through the omniview images convey enough information to compensate for the accumulation of error in odometry. Features like walls, hallways, doors and even the small "box-like" structure where the robot goes around are now clearly visible.

Regarding the representation of the information matrix, the diagonal structure discussed in section III can be appreciated (see figure 6, Left). The non diagonal elements represent the information introduced when a loop is closed. The small non diagonal elements crossing perpendicularly the main diagonal represent the information introduced when small loops are closed. The other non diagonal elements shown far away from the main diagonal but in the same direction, represent the information introduced when the big loop is closed and the robot comes back to the initial position. It is important to note the clear difference between the information matrix and the error covariance matrix. In the information matrix, new information is only introduced as links between the previous pose and the following one and only additional information is present in the case of loop closure. On the other hand (see figure 6, Right), the correlations present in the error covariance matrix are updated at every stage and the matrix presents a "checkers board-like" structure. The fact that the information matrix presents so many "white" space (actual zeros in the matrix) is a fundamental gain in computational complexity as only a few elements in the matrix are updated at every step, hence the matrix is naturally sparse. [2]

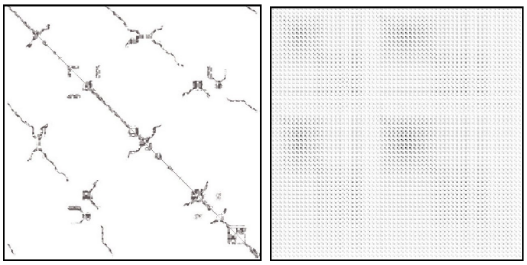

Fig. 6. **Left**: Information matrix, **Right**: error covariance matrix - small office

Regarding the computational benefits of the EIF with respect to the EKF standard solution, we present in figure 7 the computation times. This plot is only show for the small office experiments as the times required to compute the EKF on more than 1000 robot poses are very high.

C. Large Office Environment

For the large office environment two maps are presented. The first map is based on the odometry readings (see figure 8), while the second map represents the SLAM corrected trajectory (see figure 9). For this maps we do not show the laser readings as the size of the trajectory is much larger and the laser becomes unreadable.

[2]In our experiments we also run the standard EKF solution to compare obtaining indeed the exact same results.

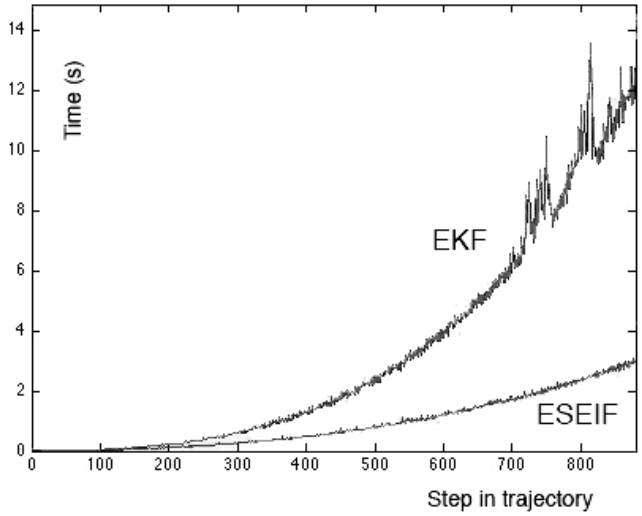

Fig. 7. Computational cost of EKF vs. EIF. Data association and feature extraction are not included.

Figure 9 shows the resulting SLAM corrected trajectory of the robot. As we can see, the large loop is correctly closed even though the accumulated error in odometry was very large. However, the resulting SLAM corrected trajectory is not as truthful as it was for the case of the small data-set. The reason for this is the amount of correction needed to close such a large gap in the loop closing point. When previously seen areas are detected, the whole trajectory needs to be *bent* in order for those areas to overlap. Given the constraints between different robot poses, the *bending* is applied over the complete trajectory, reconnecting the loop closing portions, but spreading the rest of the trajectory. This behavior in the loop closing is similar of that of a piece of wire that is bent putting both extremes together. As there are forces between the individual cells of the wire, the whole shape of the wire is modified when connecting both extremes. This behavior was also termed "Certainty of Relations despite Uncertainty of Position" by Udo Frese [23] and it is a direct result of the strong relations introduced in the trajectory by both the observations and the odometry.

In figure 10 we see the amount of matches found at every time step. Only in the loop closing points or the stationary moments the amount of observations increases. For the large loop after $5,210$ images, we see a sudden increase in the amount of matches which represents the robot driving the same hallway. As this portion of the trajectory was driven before, the amount of matches doubles after revisiting for the first time. Again, after $7,800$ images, the same hallway is visited again and the number of matches again doubles. The smaller peaks in the plot represent portions where the robot either closed small loops or the robot stood still for some seconds.

D. State Recovery

Regarding the computation time of the state recovery, we appreciate in figure 11 some interesting results. Initially,

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN: 978-88-95872-01-8
pp. 297-306

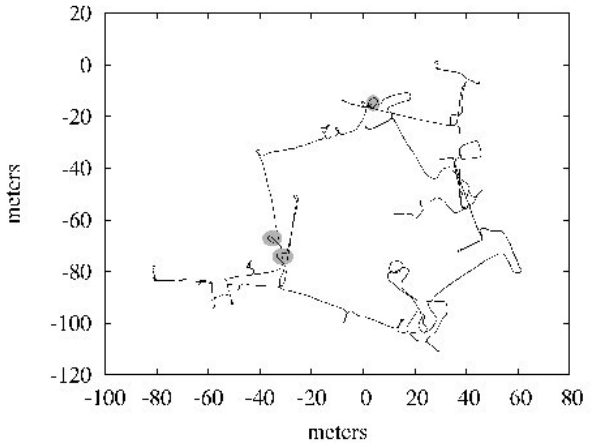

Fig. 8. Odometry based map - large loop. All grey circles represent the same spot in the trajectory

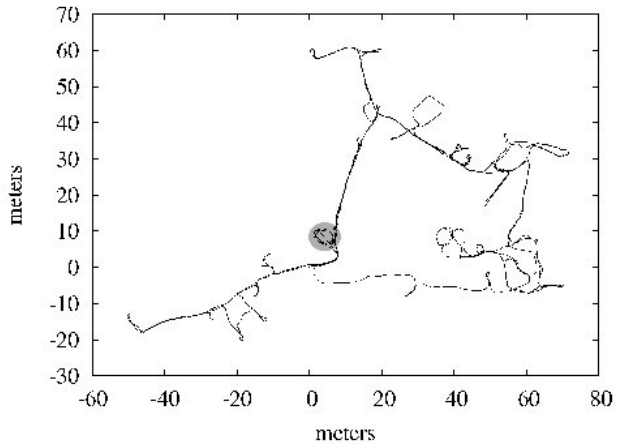

Fig. 9. SLAM based map - large loop. All grey circles represent the same spot in the trajectory

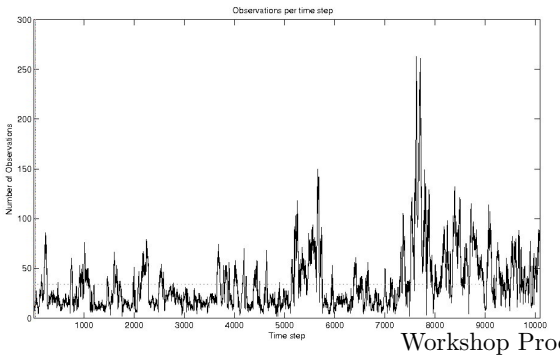

Fig. 10. Number of observations. Average 3.45

and for reasonably large data-sets, the Cholmod2 method performs better than the rest. However, this advantage in performance is actually caused by the constant number of observations. Looking closely at figures 11 and 10, we see how at the moment of the first loop closure (time state $\approx 5,210$), the growth in computation time of the Cholmod2 increases significantly and in a non linear fashion. This can be seen in figure 13 where the computation time divided by the number of non zero entries is displayed. It is clear that the CGS remains approximately constant with the number of observations. This is an interesting result key to a sucessfull implementation of the EIF over large data-sets. For an efficient implementation, a mixed strategy could be employed, using the Cholmod2 method when the number of observations is limited, and switching to CGS when loops are closed. The naive inversion technique grows so fast that it is barely useful for maps with more than 100 features. The PCG technique shows a reasonable performance up to time step $1,000$. The sudden increase in computation time could be caused by an inappropriate conditioning. Lu and CGS behave very similarly though their performance is significantly worse than Cholmod2 for a constant amount of observations. This difference in performance is specially noticeable for data-sets with more than $\approx 2,000$ robot poses. The time required by LU or CGS is more than double at time step $3,000$, which implies that the total computation time up to that time step is 27 minutes for LU and CGS and 10 minutes for Cholmod2.

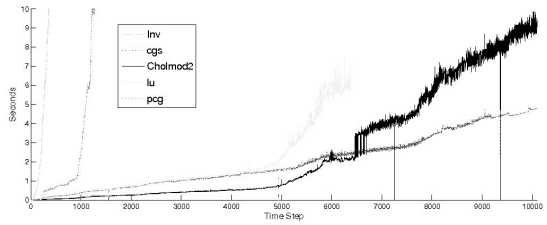

Fig. 11. Computation speed per time step in state recovery. As we use a trajectory based approach, the time step multiplied by 3 is the number of map features (number of robot poses $\times$ dimensions of each pose)

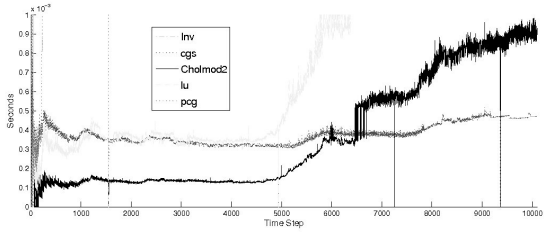

Fig. 12. Computation speed per time step in state recovery divided by the number of non zero entries in the information matrix.

As the CGS is not an exact solution, a comparison measure was needed to determine weather the CGS was overconfident or conservative in the estimation of the information matrix. Using a measure from [24], we compared the matrices obtained in CGS and Cholmod. The resulting histogram

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008,November,3-4
ISBN 978-88-95872-01-8
pp. 297-306

shows how the estimated information matrix in CGS is conservative with respect to the exact matrix obtained by Cholmod2. We understand that a deeper understanding on how this conservative estimation affects the overall trajectory is required.

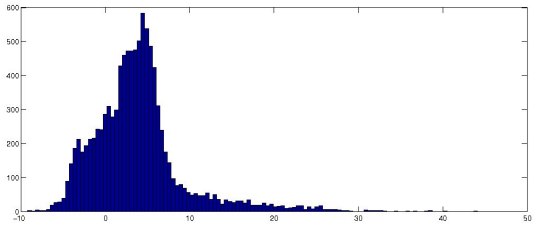

Fig. 13. Information matrix comparishon between CGS and Cholmod2.

V. Discussion and Conclusion

We have presented our Omnivideo SLAM System and we can summarize our contributions as follows. We have successfully used an omnivideo camera alone to both construct an accurate map and provide localization skills to a robot placed in a small office environment. For a larger scale environment, we have shown that the standard KF solution is not appropriate because of the computational costs. Furthermore, we have provided a comprehensive comparison between 5 different techniques to perform state recovery and have summarized and analytically derived the computational advantages of the EIF for trajectory based approaches. Regarding the data association problem and loop closure scenario, we have shown how a similarity measure can be used to detect loop closure without using priors under $O(n)$ complexity in the number of images. Finally, we hope to have shown some insight in the difficult task of accurately mapping large indoor environments with an inexpensive camera sensor where large loops are closed.

We have shown that an omnivideo system is adequate for building an accurate map of indoor office environments. Estimating the epipolar geometry of two panoramic images and obtaining the relative heading and orientation of each other yields enough information to create a consistent trajectory map and compensate for the accumulation of error in odometry. Furthermore, by means of a data-set recorded with a mobile robot, we have shown that an accurate map of a small office environment can be created and maintained by our omnivision trajectory based SEIF approach. Given the computation time required by the ESEIF and the state recovery process, we have also shown that building such a map with $\approx 1,000$ robot poses can also be done in real time.

Using a robust image matching algorithm together with the ESEIF, large loops can be effectively closed in extremely large environments. The computational gain of the Information Filter compared with the traditional Extended Kalman Filter solution showed an improvement in performance. Our large office experiment will not be possible with the standard Extended Kalman Filter due to the quadratic time required. Furthermore, it will not be possible without an appropriate state recovery technique, as we have shown that only LU decomposition, CGS and Cholmod2 are sufficiently fast for less than $5,000$ robot poses, though the Cholmod2 is significantly better in terms of global computation time. For data-sets with more than $10,000$ robot poses we show that the CGS will perform better as the computation time growth in our experiments shows a more linear behavior less sensible to number of observations. An efficient implementation will consider using the Cholmod2 for a number of robot poses below $5,000$ and a constant number of observations. When the number of features increases significantly, for instance when closing large loops, the CGS seems to be the most appropriate choice.

Having observed the amount of error induced by the linearization process by means of an artificial data set our SLAM algorithm could be improved with a more appropriate linearization technique. The essential drawback of the Information Filter is the need to recover the state in order to compute the next filter step. Some approaches regarding partial state recovery could be employed though they also introduce error as they are only an approximation. Another interesting alternative to explore will be the substitution of the non linear function $h(x)$ with an alternative function over the information vector, namely $h^*(\eta)$. This will shortcut the need to recover the state vector x though the definition of such function is difficult to foresee as the information vector lacks geometrical meaning.

Regarding the "bending" process of the trajectory on the loop closing points, we believe, that using a relaxation technique to introduce additional error between certain robot poses, the bending could be enforced over those poses, acting as joints in the bending process. This error introduction could be done by a more detailed odometry error model. By dropping the use of a static error covariance R and introducing an improved model that accounts for the additional error when the robot is turning. Such an error model will be integrated in a non-linear motion process.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I - The Essential Algorithms," *Robotics and Automation Magazine*, June 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II - State of the Art," *Robotics and Automation Magazine*, September 2006.

[3] R. Smith and P. Cheesman, "On the representation of spatial undertainty," *Robotics Research, 5(4):5668*, 1987.

[4] A.J.Davidson, I. Reid, N. Molton, and O. Stasse, "Mono-slam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.

[5] T. Vidal-Calleja, A.J.Davidson, J. Andrade-Cetto, and D. Murray, "Active control for single camera slam."

[6] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, April 2007.

[7] L. Clemente, A. Davidson, I. Reid, J. Neira, and J. Tards, "Mapping large loops with a single hand-held camera," in *Robotics: Science and Systems*, 2007.

[8] J. A. Castellanos, R. Martinez-Cantin, J. Tards, and J. Neira, "Robocentric map joining: Improving the consistency of ekf-slam," in *Robotics and Autonomous Systems*, 2007.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November, 3-4
ISBN 978-88-95872-01-8
pp. 297-306

[9] P. Newman and K. Ho, "Slam- loop closing with visually salient features," in *ICRA*, 2005.
[10] T. Svoboda and T.Pajdla, "Epipolar geometry for central catadioptric cameras," in *IJCV*, 2002.
[11] T.Goedeme, M.Nuttin, T.Tuytelaars, and L. Gool, "Omnidirectional vision based topological navigation," in *IJCV*, 2007.
[12] B. Micusik and T.Padjla, "Structure from motion with wide circular field of view cameras," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
[13] B. Kröse, O. Booij, and Z. Zivkovic, "A geometrically constrained image similarity measure for visual mapping, localization and navigation," *European Conference on Mobile Robots*, 2007.
[14] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE J. Oceanic Eng.*, 2007, Accepted, To Appear.
[15] F. Ramos, J. Nieto, and H. Durrant-Whyte, "Recognising and modelling landmarks to close loops in outdoor slam," in *ICRA*, 2007.
[16] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots, 4, 333–349*, 1997.
[17] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *IEEE Transactions on Robotics*, vol. 22, no. 6, 2006.
[18] D. K. S. Thrun, Y. Liu, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filter," *International Journal of Robotics Research*, vol. 23(7-8), 2004.
[19] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision, second edition*. Cambridge University Press, 2003.
[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[21] B. Horn, "Recovering baseline and orientation from essential matrix," January 1990, available from http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-801Fall-2004/Readings/.
[22] O. Booij, Z.Zivkovic, and B. Krose, "Sampling in image space for vision based slam," in *RSS Workshop*, 2008.
[23] U. Frese, "A discussion of simultaneous localization and mapping," *Autonomous Robots*, February 2006.
[24] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based slam," *The International Journal of Robotics Research*, February 2007.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 297-306