15

Localization and Mapping for Service Robots: Bearing-Only SLAM with an Omnicam

Christian Schlegel and Siegfried Hochdorfer
University of Applied Sciences
D-89075 Ulm
Germany

1. Introduction

Localization and mapping are fundamental problems in service robotics since representations of the environment and knowledge about the own pose significantly simplify the implementation of a series of high level applications. For instance, nearly all relevant applications of service robots require navigation skills that allow for purposeful motions. Typical examples are fetch-and-carry tasks or floor coverage tasks. These are best implemented based on pose knowledge and on continuously updated maps of the environment.

Thus, a key component towards widespread use of service robots is a localization capability that can vary from pose tracking over relocalization to even solving the most demanding so-called kidnapped robot problem. In the latter case the robot is carried to an arbitrary location during its operation and is expected to detect this and then relocalize itself.

Of course, the difficulty of the localization problem significantly depends on the available information. Normally, localization requires some kind of map as reference and map building requires pose knowledge to consistently insert artifacts. A SLAM (simultaneous localization and mapping) problem arises when the robot does neither have access to a map of the environment nor does it know its own pose. The SLAM problem is more difficult than the mapping with known poses and it is more difficult than the localization problem based on a given map.

A successful approach to overcome the chicken-and-egg problem of concurrently building a map and maintaining the robot pose is based on a probabilistic representation. The online SLAM problem maintains the robot pose and the map in a single state vector. The remaining challenge is to estimate the posterior over the current pose along with the map given all the measurements and controls. SLAM is of particular importance for service robotic applications since it significantly reduces deployment efforts and ensures continues updates as needed in dynamic environments. However, one cannot neglect the specific demands on service robots. For instance, in most applications of service robots the consumer neither accepts modifications of the environment (like artifical landmarks) nor complex and time consuming deployment efforts. Although a large body of work already proved that the SLAM problem is solvable even without deploying artifical landmarks, most approaches are based on range measuring devices. For most of the service robotics applications like floor cleaning or lawn mowing, these devices are either still too expensive (e.g. laser range finders) and do not fit into the budget or do not show the required performance (e.g. ultrasonic sensors in large open spaces like gyms or lobbies). Thus, a SLAM component based on cheap sensors requiring no artifical landmarks is a desired technology for many service robotic applications.

A solution are bearing-only SLAM approaches since these can be used with cheap sensors like omnicams. As long as no calibrated system is needed, omnicams are cheap and small and thus suitable for service robots. Although omnicams provide feature rich information on the surrounding of the robot with high update rates, they do not provide range information. Thus, one has to modify the sensor models of the well-known SLAM approaches such that observation angles of landmarks are sufficient to generate pose estimates. The problem is that one needs several observations of the same landmark from different poses to intersect the line of sights. Thus, one has to solve the so-called problem of a delayed initialization of a landmark. The problem results from the fact that the estimates of the observation poses of not yet initialized landmarks have to be corrected with each reobservation of an already known landmark. Thus, one extends the state vector such that it not only contains the robot pose and the initialized landmark poses but also the observation poses of not yet initialized landmarks.

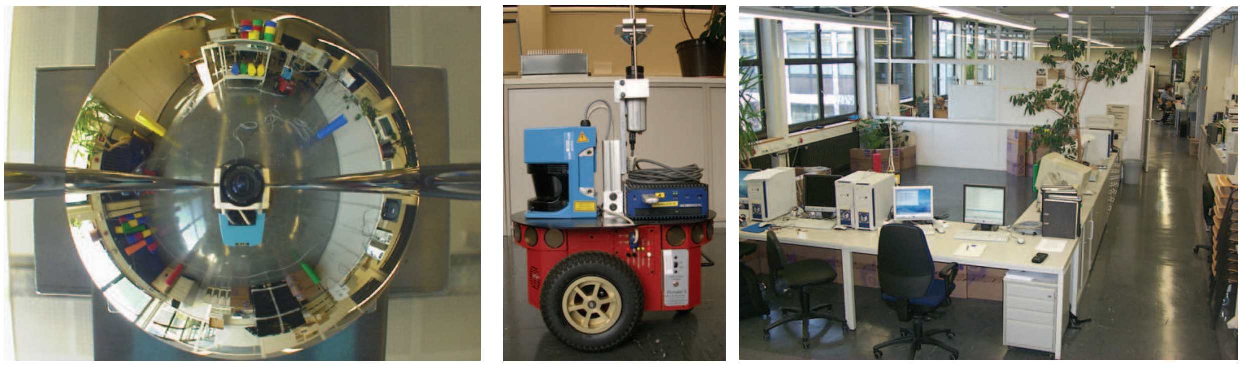


Fig. 1. The Pioneer-3DX robot with the omnicam in our everyday indoor environment.

In this chapter, we present a bearing-only SLAM system based on an omnicam. After introducing some theoretical foundations, we describe the general approach of bearing-only SLAM with an omnicam based on artificial landmarks. This setting is then extended to get rid of artificial landmarks by exploiting SIFT features (Lowe, 2004). We propose several preselection and landmark identification mechanisms that are pivotal towards the robust application of SIFT features within a bearing-only SLAM approach based on the EKF (Extended Kalman Filter). For example, exploiting viewing areas massively reduces ambiguities and mismatches in SIFT feature reobservations and thus significantly reduces false identifier assignments. The various approaches have been successfully evaluated on a Pioneer-3DX platform in a demanding indoor environment. The experimental evaluation covers characteristic requirements of advanced service robotics environments.

2. Related work

The basic idea of bearing-only SLAM with an Extended Kalman Filter (EKF) is described in (Bailey, 2003). The focus is on solving the delayed landmark initialization problem in an

EKF framework. However, the approach has been evaluated only in simulation with known landmark assignments.

The approach of Bailey is also used by Hesch & Trawny (2005). However, they apply a line filter to an omnicam image and then use vertical lines as landmarks. Again, the evaluation has been performed in simulation only.

We used the approach of Bailey as starting point for real-world evaluations of bearing-only SLAM with an omnicam. The first step was to use artificial landmarks (Schlegel & Hochdorfer, 2005). In a second step, SIFT features in an omnicam image have been evaluated for bearing-only SLAM (Hochdorfer & Schlegel, 2007). Further refinements improve the landmark validation and thus make another step towards suitability for all day use.

Other approaches for bearing-only SLAM address the problem of landmark initialization differently. Immediate initialization methods to bearing-only SLAM have also been introduced. These are called undelayed methods. Kwok et al. (2005) presented a computationally efficient multiple hypothesis approach using Gaussian sum filters to represent the initial feature state. Sola et al. (2005) then gave new insights by introducing an approach that initializes the whole vision cone. Lemaire et al. (2005) use visual features in 3D for bearing-only SLAM. They also applied an undelayed initialization method. Pros and cons of a delayed landmark initialization are discussed by Ortega et al. (2005).

Fitzgibbons & Nebot (2002) report on SLAM with a forward looking camera. The focus there is on color-based feature tracking. Davison et al. (2007) perform 3D SLAM based on a single freely-moving camera. It is also based on a standard full covariance EKF and shows impressive performance.

The main focus of the iterative SIFT feature approach of Tamimi et al. (2006) is on reducing the enormous amount of SIFT features per image. The number of keypoints can be defined in advance and the computation time is proportional to this number. The approach has been evaluated with a SLAM system using an omnicam and a particle filter.

Another improvement of the SIFT feature calculation and an evaluation what are good image features for bearing-only SLAM is presented by Wang & Zhang (2006).

An extensive experimental comparison of state-of-the art techniques for the online bearing-only SLAM problem has been performed by Bekris et al. (2006).

Meanwhile, SIFT features are widely used for SLAM problems. For example, Gil et al. (2006) use SIFT features with a stereo vision system where depth information is available with each feature. However, it is not bearing-only SLAM.

3. SLAM with an extended Kalman filter

3.1 The Kalman filter

This section gives a brief introduction into probabilistic mapping and localization. The basic model is subsequently extended for use with bearing-only SLAM. The presentation is based on (Thrun, 2005).

The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator. The Kalman filter is optimal in the sense that it minimizes the estimated error covariance given some presumed conditions are met.

The discrete Kalman filter estimates the n-dimensional state x of a discrete-time controlled process. There is typically a *process model* that models the transformation of the process state. This can usually be represented as a linear stochastic difference equation.

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{1}$$

The $n \times n$ matrix A_t describes how the state x evolves from $t-1$ to t without controls or noise. The $n \times l$ matrix B_t relates the optional control input u_t of dimension *l* to the state x. In addition there is a *measurement model* that describes the relationship between the process state and the measurements. This can usually be represented with a linear expression.

$$z_t = C_t x_t + \delta_t \tag{2}$$

The $m \times n$ matrix C_t relates the state to the measurement z_t. Measurements do not have to be of elements of the state but can be any linear combination of the state elements. The random variables ε_t and δ_t represent the process and measurement noise. They are assumed to be independent of each other and normally distributed with zero mean and covariances Q_t and R_t respectively.

The Kalman filter estimates a process by using a form of feedback control. It estimates the process state at some time and obtains feedback in the form of measurements. The *prediction step* projects forward (in time) the current state estimate and the error covariance to obtain the *a priori* estimate for the next time step. The *correction step* is responsible for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. Now one can run an ongoing cycle of prediction and correction steps. This recursive nature of the Kalman filter is one of its very appealing features that makes its practical implementation feasible.

The prediction step calculates the *a priori* estimate and the corresponding covariance matrix at time step t according to the following equations. Based on the *a posteriori* estimate of the previous time step $t-1$, the state transformation matrix and the control input, the *a priori* estimate of time step t is determined.

$$\hat{x}_t^- = A_t \hat{x}_{t-1} + B_t u_t \tag{3}$$

$$P_t^- = A_t P_{t-1} A_t^T + Q_t \tag{4}$$

The measurement step incorporates new observations. The first task is to compute the Kalman gain K_t. The Kalman gain sorts out the weights of the measurement and the prediction. It depends on the error covariance of the prediction P_t^- and the measurement error covariance R_t. The measurement and the prediction are then combined in a weighted manner to form the *a posteriori* estimate. Finally, the error covariance matrix belonging to the new estimate is calculated.

$$K_t = P_t^- C_t^T \left(C_t P_t^- C_t^T + R_t \right)^{-1} \tag{5}$$

$$\hat{x}_t = \hat{x}_t^- + K_t \left(z_t - C_t \hat{x}_t^- \right) \tag{6}$$

$$P_t = \left(I - K_t C_t\right) P_t^- \tag{7}$$

3.2 The extended Kalman filter

We so far considered a discrete-time controlled process that is governed by a *linear* stochastic difference equation. However, most realistic robotic problems involve non-linear functions in the process model and/or the measurement model.

$$x_t = g\left(u_t, x_{t-1}\right) \tag{8}$$

$$z_t = h\left(x_t\right) \tag{9}$$

The general approach is a first order Taylor series expansion of the process model and the measurement model. A Kalman filter that linearizes about the current mean is referred to as an *Extended Kalman Filter (EKF)*.

The first order Taylor series expansion of the process model is given by:

$$g(u_t, x_{t-1}) \approx g(u_t, \hat{x}_{t-1}) + \frac{\partial g(u_t, \hat{x}_{t-1})}{\partial x_{t-1}} (x_{t-1} - \hat{x}_{t-1}) \tag{10}$$

$$g(u_t, x_{t-1}) \approx g(u_t, \hat{x}_{t-1}) + G_t\ (x_{t-1} - \hat{x}_{t-1}) \tag{11}$$

The measurement model is linearized as follows:

$$h(x_t) \approx h(\hat{x}_t^-) + \frac{\partial h(\hat{x}_t^-)}{\partial x_t} (x_t - \hat{x}_t^-) \tag{12}$$

$$h(x_t) \approx h(\hat{x}_t^-) + H_t\ (x_t - \hat{x}_t^-) \tag{13}$$

The resulting complete set of EKF equations now require the Jacobians G and H. The prediction step is given by the following equations:

$$\hat{x}_t^- = g\left(\hat{x}_{t-1}, u_t\right) \tag{14}$$

$$P_t^- = G_t P_{t-1} G_t^T + Q_t \tag{15}$$

The correction step is calculated according to the following equations:

$$K_t = P_t^- H_t^T \left(H_t P_t^- H_t^T + R_t\right)^{-1} \tag{16}$$

$$\hat{x}_t = \hat{x}_t^- + K_t\left(z_t - h\left(\hat{x}_k^-\right)\right) \tag{17}$$

$$P_t = \left(I - K_t H_t\right) P_t^- \tag{18}$$

3.3 The EKF and the SLAM problem

The Extended Kalman Filter can now be used to solve the SLAM problem. The seminal idea is to have a state vector comprising both the robot pose and the landmark poses. Now the overall state can be estimated at once. Thus, the chicken-and-egg problem is solved. In case of a robot pose $(x,y,\phi)^T$ and a map of N landmarks $(x,y)^T$, we have a $(3+2N)$ dimensional Gaussian to represent the state of the SLAM problem.

The online SLAM problem estimates the posterior over the current pose along with the map:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) \tag{19}$$

The robot pose at time t is denoted x_t, m is the map, and $z_{1:t}$ and $u_{1:t}$ are the measurements and controls, respectively. Most algorithms for the online SLAM problem are incremental that is they discard past measurements and controls once these have been processed. Maps in EKF based SLAM approaches are feature-based. A map feature is usually represented as a point landmark m_i with coordinates x_i and y_i.

The Kalman filter consists of a prediction and a correction step executed in a loop. For the SLAM problem, the prediction step is based on a motion model of the robot taking into account the control inputs. The correction step integrates landmark observations and is based on an observation model. The observation model relates the state vector to the measurement, that is, it allows to calculate the expected measurement.

4. Bearing-only SLAM

4.1 The state vector

Bearing-only SLAM requires several observations of the same landmark from different robot poses to intersect line of sights to calculate a landmark pose. Since a new landmark can be initialized only after collecting a bunch of measurements, we call this a delayed landmark initialization. The problem now is that while collecting several observations, the robot pose already gets updated by the SLAM mechanism. Since the various observation poses are not independent of each other, these need to be updated as well. Otherwise, the lateron performed intersection of line of sights from different observation poses would not be consistent anymore. The basic idea now is to include the observation poses into the SLAM state vector. Now these observation poses are consistently updated by the SLAM mechanism. Since the measurements are relative to the observation poses and since these are updated consistently, we can consistently transform angular observations into line of sights in the global frame of reference even when the measurements are from different points in time. Thus, a deferred but consistent evaluation of measurements is possible. Measurements of a possible landmark can now be accumulated over time until sufficient information is available for a reliable landmark initialization.

The extended state vector for bearing-only SLAM with a delayed landmark initialization is thus given by (Bailey, 2003):

$$x = \left[x_v^T, x_{v_m}^T, \ldots, x_{v_1}^T, x_{f_1}^T, \ldots, x_{f_n}^T\right]^T \tag{20}$$

The vehicle pose $x_v = \left[x_v, y_v, \phi_v\right]^T$ is the position and heading of the robot in the global frame of reference. The entry $x_{v_i} = \left[x_{v_i}, y_{v_i}, \phi_{v_i}\right]^T$ represents an observation pose where not yet evaluated measurements are available. Each pose x_{v_i} corresponds to the time and location where a set of measurements $\left\{\theta_{v_i}^1, \ldots, \theta_{v_i}^k\right\}$ was obtained. Already initialized landmarks $x_{f_i} = \left[x_{f_i}, y_{f_i}\right]^T$ are represented by their 2D position without orientation.

4.2 The overall processing scheme

Before describing each step of the bearing-only SLAM approach in detail, we first give a short overview on the overall processing scheme. The processing order per time-step executing the loop of correction and prediction steps of the EKF is as follows (Bailey, 2003):

- Perform prediction step according to motion model.
- Get image from omnicam and get robot pose from odometry. Next, we extract features and determine angle to features.
- Measurements of existing map features are processed first in a batch update.
- If there exists a well-conditioned pair of measurements for a non-initialized landmark, the initial landmark estimate is added to the SLAM state vector.
- For each newly initialized feature, the remaining accumulated measurements are applied in a batch update.
- Observation poses x_{v_i} are removed from the SLAM state vector as soon as they do no longer possess not yet processed measurements. In that case these observation poses are not needed any longer.
- In case the current robot pose provides a measurement for a not yet initialized landmark, the observation is stored and the current robot pose is added as new observation pose to the SLAM state vector by *stochastic cloning* (Roumeliotis & Burdick, 2002):

$$\begin{bmatrix} \hat{x}_v^T \\ \hat{x}_m^T \end{bmatrix} \rightarrow \begin{bmatrix} \hat{x}_v^T \\ \hat{x}_v^T \\ \hat{x}_m^T \end{bmatrix}, \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix} \rightarrow \begin{bmatrix} P_{vv} & P_{vv} & P_{vm} \\ P_{vv} & P_{vv} & P_{vm} \\ P_{vm}^T & P_{vm}^T & P_{mm} \end{bmatrix} \tag{21}$$

4.3 The motion model for the Kalman filter

The motion model describes the pose change of the robot given the current pose and control commands. Since action execution is always subject to errors, the motion model is used with stochastic variables. The motion model then propagates the pose estimate according to the motion model and given control commands and adds the additional uncertainty representing flawed execution.

As can be seen in figure 2, the pose x_{k+1} is reached from pose x_k by applying the virtual control command $\left(d_1, \Delta\alpha_1, \Delta\alpha_2\right)$. This virtual control command is calculated out of two odometry reports x_{k+1} and x_k. Thus, we regularly sample the odometry values and approximate each motion between two poses by first turning towards the new pose x_{k+1} by $\Delta\alpha_1$, then driving the distance d and finally adjusting the heading by a final turn $\Delta\alpha_2$. The motion model can now be used to carry forward covariances of pose estimates based on two subsequent odometry reports. This model always considers the total amount of rotation even for S-shaped trajectories. This is important since rotations induce high uncertainties on the pose estimate of the robot. It now depends on the odometry sampling rate on how tight the approximation (blue line) captures the actual motion (red line).

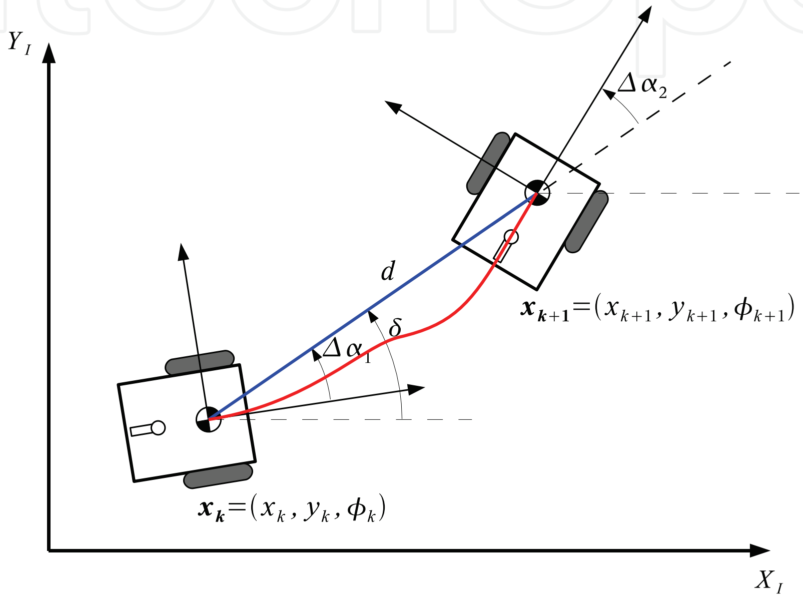

Y_I
$\Delta\alpha_2$
d
δ
$\Delta\alpha_1$
$\boldsymbol{x_{k+1}}=(x_{k+1}, y_{k+1}, \phi_{k+1})$
$\boldsymbol{x_k}=(x_k, y_k, \phi_k)$
X_I

Fig. 2. The robot motion model.

The full motion model is given by the following equation:

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{pmatrix} = m(x_k, y_k, \phi_k, d, \Delta a_{1,} \Delta a_{2,} \Delta\beta) = \begin{pmatrix} x_k + d\cos(\phi_k + \Delta a_1) \\ y_k + d\sin(\phi_k + \Delta a_1) \\ \phi_k + \Delta a_1 + \Delta a_2 + \Delta\beta \end{pmatrix} \quad (22)$$

The parameters of the motion model are calculated as follows with $\Delta\beta$ a drift error that depends on the driving distance:

$$d = \sqrt{(y_{k+1} - y_k)^2 + (x_{k+1} - x_k)^2} \quad (23)$$

$$\delta = arctan\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) \quad (24)$$

$$\Delta a_1 = \delta - \phi_k \quad (25)$$

$$\Delta a_2 = \phi_{k+1} - \delta \tag{26}$$

$$\Delta\beta = s(d) \tag{27}$$

The Jacobian, which is needed to propagate the covariances in the linearized motion model of the EKF, is given by

$$\frac{\partial m(\dots)}{\partial\left(x_k, y_k, \varphi_k, d, \Delta\alpha_{1,} \Delta\alpha_{2,} \Delta\beta\right)} = \begin{bmatrix} 1 & 0 & -d\,sin\left(\varphi_k + \Delta\alpha_1\right) & cos\left(\varphi_k + \Delta\alpha_1\right) & -d\,sin\left(\varphi_k + \Delta\alpha_1\right) & 0 & 0 \\ 0 & 1 & d\,cos\left(\varphi_k + \Delta\alpha_1\right) & sin\left(\varphi_k + \Delta\alpha_1\right) & d\,cos\left(\varphi_k + \Delta\alpha_1\right) & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{28}$$

The variances of the control commands are given by $\sigma_d^2 = |d|\lambda_d$ with $\lambda_d\left[mm^2/mm\right]$, $\sigma_{\Delta\alpha}^2 = |\Delta\alpha|\lambda_{\Delta\alpha}$ with $\lambda_\alpha\left[rad^2/rad\right]$ and $\sigma_{\Delta\beta}^2 = |d|\lambda_{\Delta\beta}$ with $\lambda_\beta\left[rad^2/mm\right]$.

4.4 Angular measurements in omnicam images

An omnicam image maps the environment radially. The center point corresponds to the optical axis of the camera. It is important to note that we only use the observation angles of landmarks. The observation angle of a landmark in the omnicam image is equivalent to the yaw-angle γ of the landmark in the 3d-environment when using polar-coordinates (pitch β, yaw γ, distance d). As can be seen in figure 3, the image distortion of the omnicam does not affect the observation angle. Of course, all landmarks with the same yaw-angle γ but different pitch-angles β possess the same observation angle for the SLAM procedure.

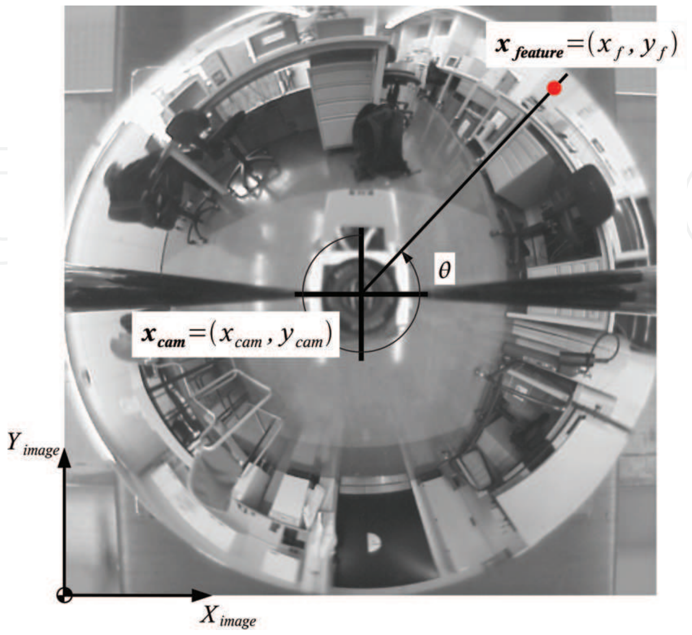


Fig. 3. Calculation of the observation angle of a feature in an omnicam image.

The calculation of the observation angle can be done in the image coordinate system of the omnicam image. The optical axis of the camera in image coordinates is denoted by x_{cam} and the feature position in image coordinates by $x_{feature}$. The observation angle relative to the robot is denoted by $z_{observation\ angle}$.

$$z_{observation\ angle} = h\left(x_{cam}, x_{feature}\right) = arctan\left(\frac{y_{feature} - y_{cam}}{x_{feature} - x_{cam}}\right) + n_i \tag{29}$$

The measurement error n_i is modeled as zero-mean white gaussian noise with variance σ_θ^2 . The variance is set to a fixed value independent of the image coordinates of the landmark. This approach assigns the same worst case uncertainty to all landmark observation angles. We assume a one-pixel jitter of a landmark position in image coordinates. The closer the landmark is to the center point, the bigger is the effect of the one-pixel jitter on the angular error. However, the minimum distance of landmarks to the center point is restricted due to the characteristics of the omnicam. The center part of the image contains the robot itself and thus is masked out for potential landmarks.

4.5 The observation model for the Kalman filter

The observation model of a landmark is used to integrate another observation of an already initialized landmark by means of the Kalman filter. The Kalman filter exploits the difference between the expected angular measurement and the current measurement. Instead of image coordinates, we now use the current robot pose estimate x_v and the current pose estimate of the considered landmark x_{f_i} . Of course, we now have to take into account the robot heading ϕ_v . The observation model can be expressed by

$$z_i = h\left(x_v, x_{f_i}\right) = arctan\left(\frac{y_{f_i} - y_v}{x_{f_i} - x_v}\right) - \phi_v + n_i \tag{30}$$

with n_i the measurement error modeled as zero-mean white gaussian noise with variance σ_θ^2 . Again, a fixed value representing the worst case is used for all landmark angles.

The EKF update equations require the Jacobian of the observation model with d_{v,f_i} the euclidean distance between vehicle and landmark. The position of H_{f_i} in H corresponds to the position of the landmark in the state vector.

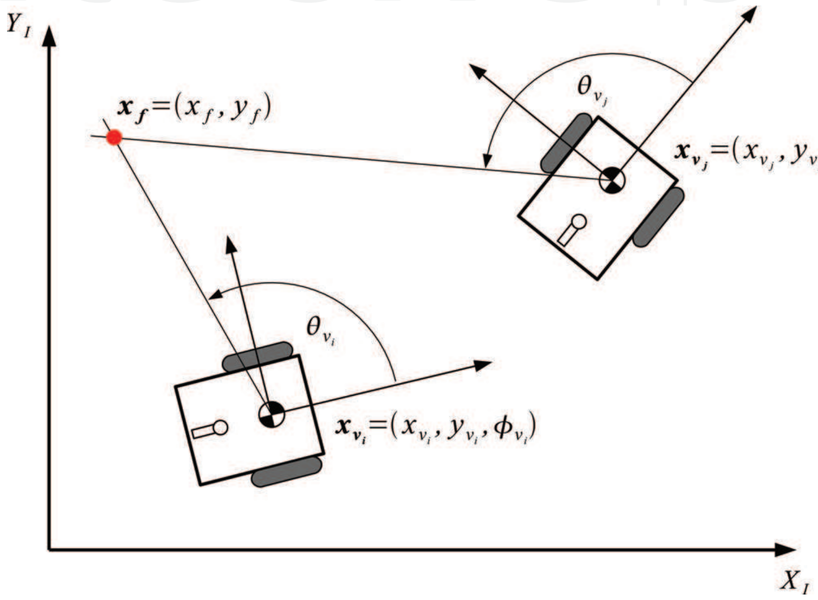
$$H = \nabla_x h(\hat{x}) = \left\lfloor H_v \quad 0 \ldots 0 \quad \mathrm{H}_{f_i} \quad 0 \ldots 0 \right\rfloor \tag{31}$$

$$H_v = \left[\frac{\partial h}{\partial \hat{x}_v} \quad \frac{\partial h}{\partial \hat{y}_v} \quad \frac{\partial h}{\partial \hat{\phi}_v}\right] = \left[\frac{\hat{y}_{f_i} - \hat{y}_v}{\hat{d}_{v,f_i}^2} \quad -\frac{\hat{x}_{f_i} - \hat{x}_v}{\hat{d}_{v,f_i}^2} \quad -1\right] \tag{32}$$

$$H_{f_i} = \begin{bmatrix} \dfrac{\partial h}{\partial \hat{x}_{f_i}} & \dfrac{\partial h}{\partial \hat{y}_{f_i}} \end{bmatrix} = \begin{bmatrix} -\dfrac{\hat{y}_{f_i} - \hat{y}_v}{\hat{d}^2_{v,f_i}} & \dfrac{\hat{x}_{f_i} - \hat{x}_v}{\hat{d}^2_{v,f_i}} \end{bmatrix} \tag{33}$$

4.6 Landmark initialization

For proper initialization of a landmark in SLAM with bearing-only information, at least two bearing measurements z_i and z_j from two different vehicle poses x_{v_i} and x_{v_j} are needed. In case of no errors, the true location of the landmark would then be given by intersecting two lines given in point-slope form as illustrated in figure 4.



Y_I
$x_f=(x_f, y_f)$
θ_{v_j}
$x_{v_j}=(x_{v_j}, y_{v_j}, \phi_{v_j})$
θ_{v_i}
$x_{v_i}=(x_{v_i}, y_{v_i}, \phi_{v_i})$
X_I

Fig. 4. Calculation of the landmark position based on two angular observations.

Given the following point-slope form

$$\begin{aligned} \left(y_f - y_{v_i}\right) &= \tan\left(\theta_i + \phi_{v_i}\right)\left(x_f - x_{v_i}\right) \\ \left(y_f - y_{v_j}\right) &= \tan\left(\theta_j + \phi_{v_j}\right)\left(x_f - x_{v_j}\right) \end{aligned} \tag{34}$$

the intersection is obtained by:

$$\begin{aligned} x_f &= \begin{bmatrix} x_f \\ y_f \end{bmatrix} = g\left(x_{v_i}, x_{v_j}, \theta_i, \theta_j\right) \\ &= \begin{bmatrix} g_x(\ldots) \\ g_y(\ldots) \end{bmatrix} = \begin{bmatrix} \dfrac{x_{v_i} s_i c_j - x_{v_j} s_j c_i + \left(y_{v_j} - y_{v_i}\right) c_i c_j}{s_i c_j - s_j c_i} \\ \dfrac{y_{v_j} s_i c_j - y_{v_i} s_j c_i + \left(x_{v_i} - x_{v_j}\right) s_i s_j}{s_i c_j - s_j c_i} \end{bmatrix} \end{aligned} \tag{35}$$

where we abbreviate $s_i = sin\left(\phi_{v_i} + \theta_i\right)$ and $c_i = cos\left(\phi_{v_i} + \theta_i\right)$.

In case of noise-corrupted vehicle poses and measurements, the landmark estimate is given by

$$\begin{bmatrix} \hat{x}_f \\ \hat{y}_f \end{bmatrix} = g\left(\hat{x}_{v_i}, \hat{x}_{v_j}, z_i, z_j\right) \tag{36}$$

with $z_i = \theta_i + n_i$ and $z_j = \theta_j + n_j$ where n_i and n_j denote zero mean and white gaussian noise. Using first order Taylor series expansion for $g(\ldots)$ and taking into account that the measurement noise is independent of the vehicle pose, one can approximate the 2×2 covariance matrix P_{LL} of the landmark position estimate by

$$P_{LL} = GPG^T + W\begin{pmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix}W^T \tag{37}$$

and the correlation with the other entries in the SLAM state vector, P_{LX} and P_{XL}, by $P_{LX} = P_{XL}^T = HP$ assuming uncorrelatedness between system and measurement error (Hesch & Trawny, 2005) where $t = \left(s_i c_j - s_j c_i\right)^2$ and

$$G = \nabla_x g(\hat{x}) = \begin{bmatrix} G_{v_i} & 0\ldots0 & \mathrm{G}_{v_j} & 0\ldots0 \end{bmatrix} \tag{38}$$

$$G_{v_i} = \nabla_{x_{v_i}} g(\hat{x}) = \begin{pmatrix} \dfrac{\partial g_x}{\partial x_{v_i}} & \dfrac{\partial g_x}{\partial y_{v_i}} & \dfrac{\partial g_x}{\partial \varphi_{v_i}} \\ \dfrac{\partial g_y}{\partial x_{v_i}} & \dfrac{\partial g_y}{\partial y_{v_i}} & \dfrac{\partial g_y}{\partial \varphi_{v_i}} \end{pmatrix}$$
$$= \frac{1}{t}\begin{pmatrix} s_i c_j\left(s_i c_j - s_j c_i\right); & -c_i c_j\left(s_i c_j - s_j c_i\right); & \left(\hat{x}_{v_j} - \hat{x}_{v_i}\right)s_j c_j - \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)c_j^2 \\ s_i s_j\left(s_i c_j - s_j c_i\right); & -s_j c_i\left(s_i c_j - s_j c_i\right); & \left(\hat{x}_{v_j} - \hat{x}_{v_i}\right)s_j^2 - \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)s_j c_j \end{pmatrix} \tag{39}$$

$$G_{v_j} = \nabla_{x_{v_j}} g(\hat{x})$$
$$= \frac{1}{t}\begin{pmatrix} -s_i c_j\left(s_i c_j - s_j c_i\right); & -c_i c_j\left(s_i c_j - s_j c_i\right); & \left(\hat{x}_{v_i} - \hat{x}_{v_j}\right)s_i c_i + \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)c_i^2 \\ -s_i s_j\left(s_i c_j - s_j c_i\right); & -s_j c_i\left(s_i c_j - s_j c_i\right); & \left(\hat{x}_{v_i} - \hat{x}_{v_j}\right)s_i^2 + \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)s_i c_i \end{pmatrix} \tag{40}$$

$$W = \nabla_n g(\hat{x})$$
$$= \frac{1}{t}\begin{pmatrix} \left(\hat{x}_{v_j} - \hat{x}_{v_i}\right)s_j c_j - \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)c_j^2; & \left(\hat{x}_{v_i} - \hat{x}_{v_j}\right)s_i c_i + \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)c_i^2 \\ \left(\hat{x}_{v_j} - \hat{x}_{v_i}\right)s_j^2 - \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)s_j c_j; & \left(\hat{x}_{v_i} - \hat{x}_{v_j}\right)s_i^2 + \left(\hat{y}_{v_j} - \hat{y}_{v_i}\right)s_i c_i \end{pmatrix} \tag{41}$$

4.7 Deciding on landmark initialization

The above described landmark initialization can only be applied if suitable observations are available. An initialization is performed only if there exists a well-conditioned pair of measurements for a non-initialized landmark. As described in detail in (Bailey, 2003), "a new feature is considered well-conditioned if the true probability density function of its location closely resembles the Gaussian approximation obtained from a Jacobian-based linearized transform".

The initial landmark pose estimate is calculated by intersecting line of sights. This calculation includes non-linear operations. Thus, the calculation of the covariance of the initial landmark estimate is not straightforward. The standard approach is to use the above derived linearizations of the calculation of the intersection point. This allows to transform the covariances of the observation poses and of the angular measurements into a covariance of the initial landmark estimate. However, the result might deviate substantially from the true distribution due to the effects of linearization. Since the Kalman filter is extremely brittle with respect to such inconsistencies, we have to introduce some kind of "quality measure" that only selects well-conditioned pairs of measurements.

For this, the difference between the linearized and the non-linearized calculation is considered. In short, the algebraic density transformation of the 8-D Gaussian distribution $p(x_{v_i}, x_{v_j}, \theta_{v_i}, \theta_{v_j})$ into the 8-D probability density function $p(x_{v_i}, x_{v_j}, x_f)$ is emulated by sampling. Each sample is transformed to a landmark pose according to the equation for calculating the intersection point. As result, a set of samples representing the non-Gaussian distribution of the landmark estimate is available. Each sample is scaled by the normalization factor of the algebraic density transformation evaluated at its sample value. This is the weight w_k. Furthermore, each sample gets another weight v_k based on the Gaussian of the linearized transformation. The sample relative entropy can now be calculated based on both weights by $\frac{1}{n}\sum_{k=1}^{n}\left(\ln w_k - \ln v_k\right)$. This is an approximation of the *Kullback-Leibler-distance* (Cover & Thomas, 1991). The smaller the relative entropy value is, the more similar are both probability density functions. Of course, this is only a heuristic to give a hint on whether a pair of measurements results in an initial landmark estimate that does not suffer too much from linearization effects.

5. Bearing-only SLAM with artificial landmarks

5.1 Feature extraction with artificial landmarks

The color segmentation is performed in HSV color space. The segmentation is based on static intervals per predefined landmark color for the hue and saturation channels. A color blob is identified as landmark if its size and its compactness values fall into a predefined interval. The angle is defined by the polar line starting in the center of the image and going through the color blob such that the lateral error of the color blob boundary with respect to the line is minimal. Of course, the data association problem cannot be solved in case of nearby landmarks of the same color. However, for evaluation purposes of the overall bearing-only SLAM approach, we manually placed landmarks such that different landmarks of the same color never appear close to each other in the omnicam image.

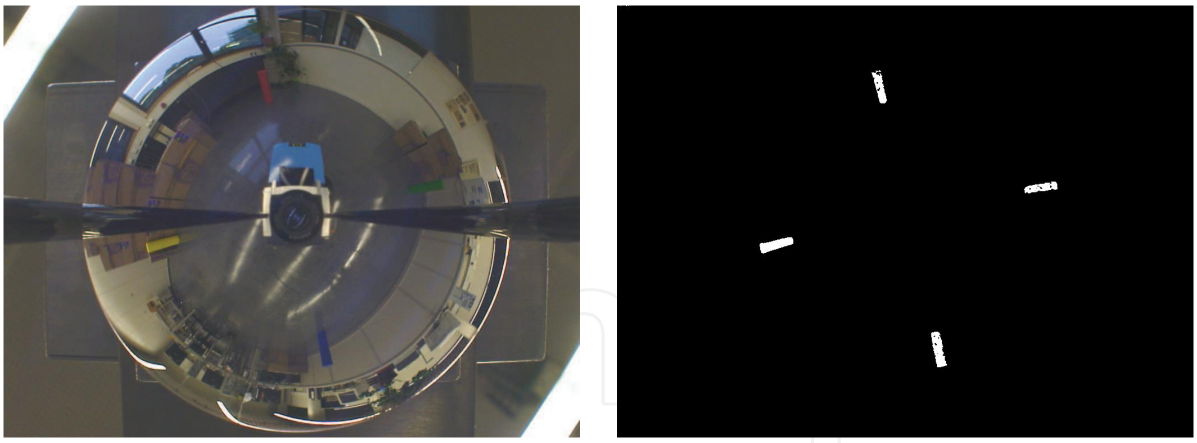


Fig. 5. The left image shows the omnicam image with colored cylinders as artificial landmarks. The right image shows the result after the segmentation step.

5.2 Experimental setup

To perform the experiments on a real world platform, we implemented all necessary steps such that a closed loop system including a Pioneer-3DX robot, an omnicam and the bearing-only SLAM algorithm is available. The omnicam is a Sony DFW-X710 camera (1024x768, 1/3 inch, progressive scan, firewire, YUV color, 15 images/second) with a hyperbolic glass mirror (H3G, Neovision). The robot is equipped with an on-board Pentium M class small sized PC. The software is Linux based and is a combination of C++ code with calls to a Matlab server for the SLAM algorithm. The image processing is based on the *Open Source Computer Vision Lib* (Intel, 2008).

The parameters of the motion model are $\lambda_d = (0,05m)^2 / 1m$ (distance error), $\lambda_\alpha = (5\deg)^2 / 360\deg$ (rotational error) and no drift error. The sensor model uses $\sigma_\theta^2 = (1.5\deg)^2$ as angular error of the landmark detection independent of the image coordinates of the landmark. This value is set larger than the theoretical one-pixel error since the segmentation process introduces further jitter on the object boundary with effects on the landmark center point. The threshold of the distance metric to decide on the initial integration of a landmark is set to 12. This threshold has been determined empirically. We set the threshold such that the angle enclosed by two line of sights of a well-conditioned pair of measurements is not below a minimum angle. This typically avoids ill-formed Gaussians for the landmark pose estimate. The robot typically moved about 1m between measurements.

5.3 Experimental results

The environment of the loop closure experiment is shown in figure 6. Figure 7 shows the sensing steps 1, 4, 5, 14, 23, 28 and 31 of a 36 step run. The ellipses show the 2-sigma contour. All units are meter. The map rotates since there is no absolute initial reference. The green lines indicate which landmark was seen by the robot in that step. The bottom right figure shows the standard error over all relative landmark distances. Since the landmarks are static, we can easily determine the ground truth relative distance between each pair of landmarks. The relative distances can also be calculated based on the landmark poses estimated by the SLAM approach. The squared error between the actual and the estimated distances is then summed up and plotted over time steps. Of course, the overall error grows by including new landmarks since more relations between landmarks are covered. However, as long as the number of landmarks is not extended, the correction steps reduce the overall error in the landmark relationships and thus improve the overall consistency of the landmark poses.

The experiments on a real platform prove that EKF-based bearing-only SLAM methods can be applied to features extracted from an uncalibrated omnicam. The 2-sigma value of the robot pose error is typically 15cm. However, artificial landmarks are not suitable for most applications and we thus have to get rid of them.

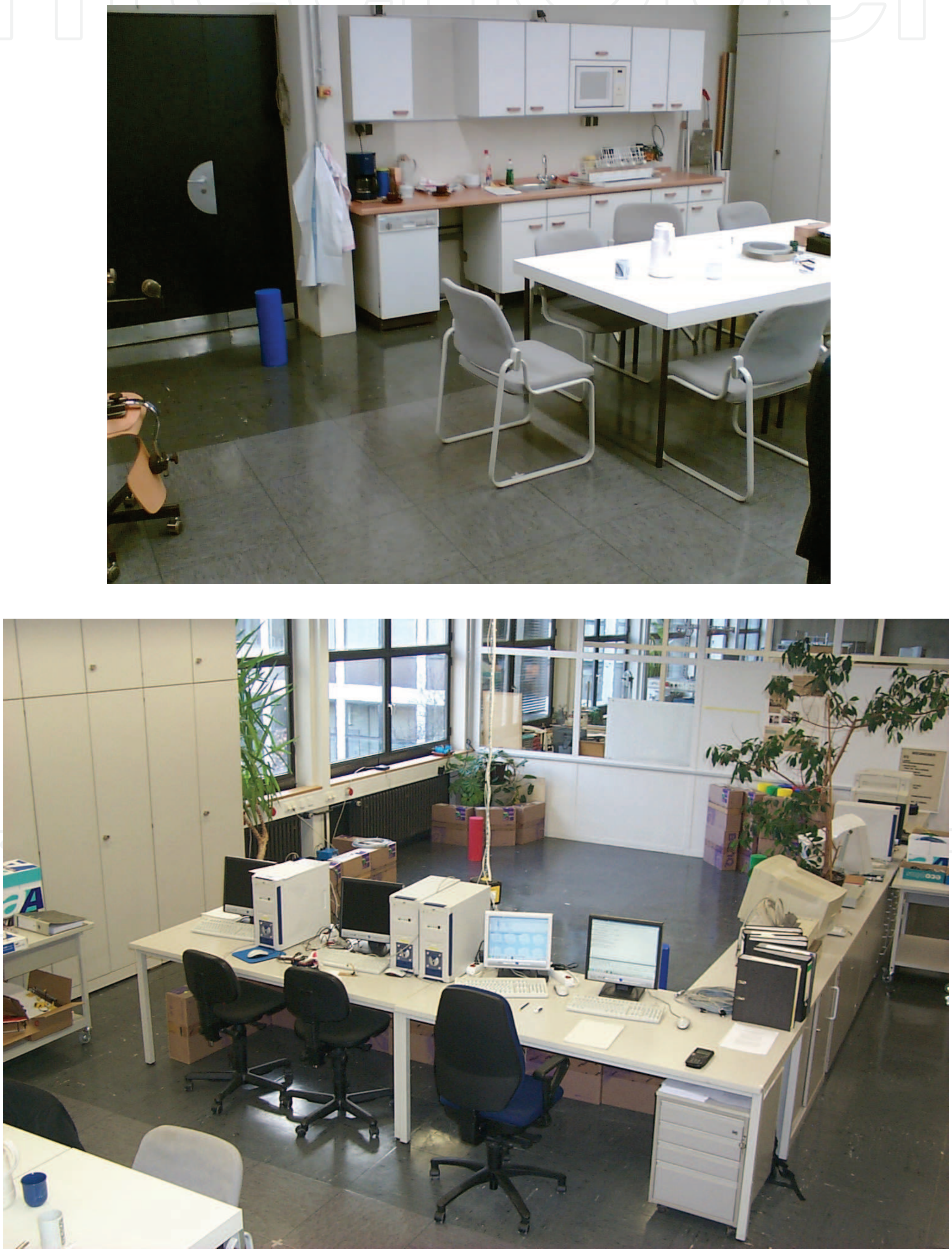


Fig. 6. The everyday indoor lab environment where this experiment has been performed.

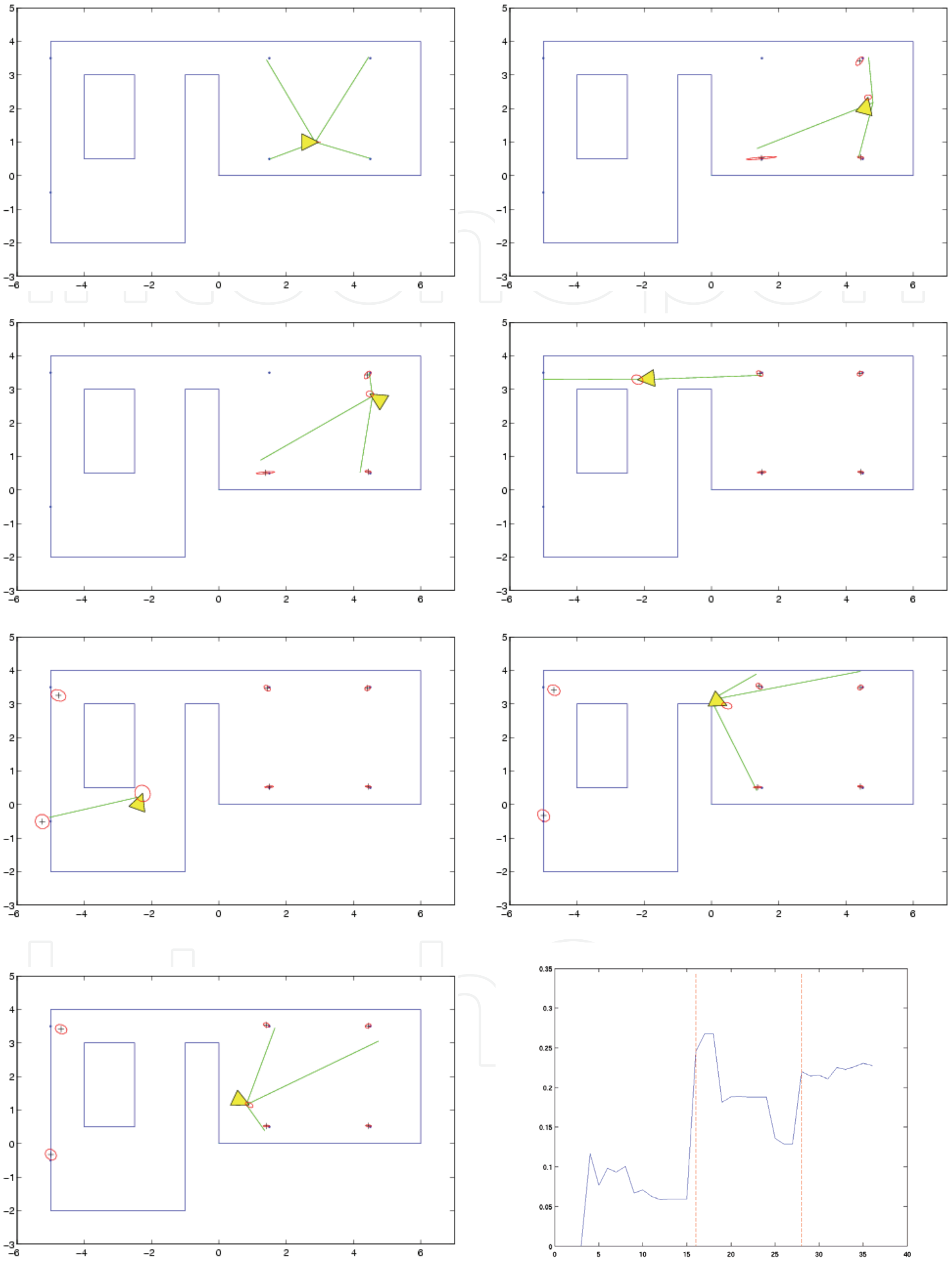


Fig. 7. The sensing steps 1, 4, 5, 14, 23, 28 and 31 of a 36 step run with 2-sigma contours. This experiment is based on artificial landmarks.

6. Bearing-only SLAM with SIFT features

Instead of artificial landmarks, we now consider SIFT features as natural landmarks for bearing-only SLAM. The SIFT approach (scale invariant feature transforms) takes an image and transforms it into a "large collection of local feature vectors" (Lowe, 2004). Up to a certain degree, each feature vector is invariant to scaling, rotation or translation of an image. SIFT features are also very resilient to the effects of noise in an image. For instance, we do not have to rely on specific shape or color models.

Depending on the parameter settings, a single omnicam image contains up to several hundred SIFT features. However, the Kalman filter based approach shows two characteristics that need to be addressed. It does not scale well with increasing numbers of landmarks and it is very brittle with respect to false observation assignments. Thus, one needs a very robust mechanism to select a small but stable number of SIFT features in an image. Potential landmarks have to be distributed sparsely over the image and should also possess characteristic descriptor values to avoid false assignments.

We still represent landmarks by 2-D poses as already explained in the previous section.

6.1 Calculation of SIFT features

SIFT features of an image are calculated by a four-step procedure. We apply the plain calculation scheme described in detail in (Lowe, 2004).

The first step is named *scale-space extrema detection*. The input images of the omnicam are of size 480x480 pixels. The first octave consists of five images, that is the original image and another four images. The latter are obtained by repeatedly convolving the original image with Gaussians. We use a σ-value of 2.4. This parameter is very robust and can thus be determined empirically. A larger value increases the computational load without improving the re-recognition of SIFT features. It is set such that the output of the overall processing chain is a stable set of roughly 90 SIFT features. In the next step, the four DOG (difference of Gaussians) images are calculated. Afterwards, extrema are detected in the two inner DOG images by comparing a pixel to its 26-neighbors in 3x3 regions. We use a down-sampling factor of 2 where downsampling ends at an image of 4x4 pixels. Therefore, we consider 7 octaves. The different octaves are illustrated in figure 8.

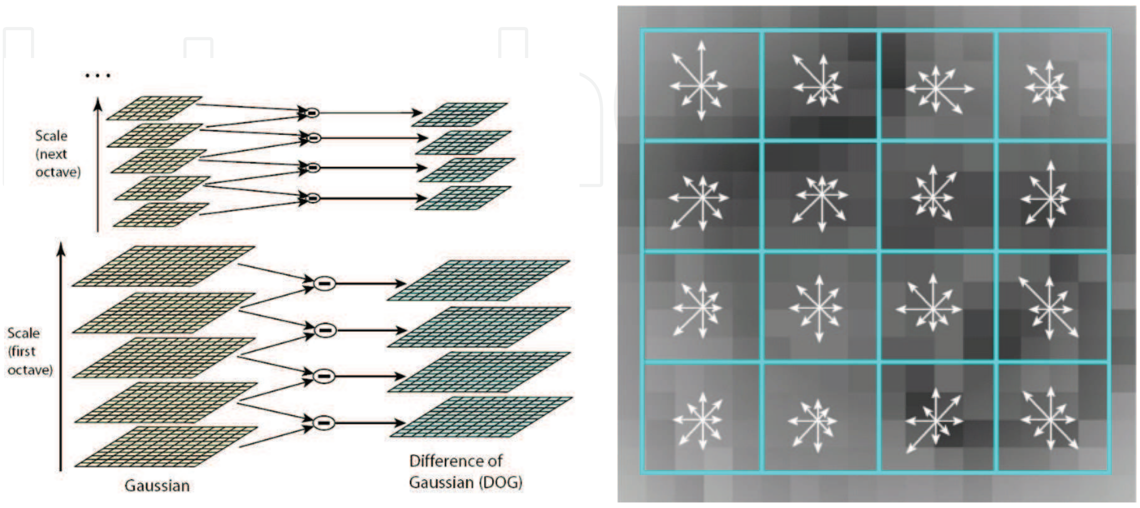


Fig. 8. The left image shows the structure of the scale space (Lowe, 2004). The right image shows the structure of a keypoint descriptor (Rihan, 2005).

The second step is named *keypoint localization*. The contrast threshold is set to 0.10. Again, the value is determined empirically. We first set it such that we obtain stable landmarks. Then we modify this threshold to reduce the number of obtained landmarks. Finally, we modify it to reduce the computational load without further reducing the number of landmarks. The curvature threshold is set to 10 (same as Lowe).
The third step is named orientation assignment. Since we do not further exploit the orientation value, we omit this step.
The fourth step is named keypoint descriptor. As described in (Lowe, 2004), we use 4x4 sample regions with 8 gradients and perform a Gaussian weighting with σ=1.5. The result are SIFT feature vectors each of dimension 128 with 8 bit entries.

6.2 The overall sequence of processing steps using SIFT features

The overall sequence of processing steps is shown in figure 9. It is important to note that we still only use the observation angles of landmarks. In case of distinct SIFT feature vectors, we just have different landmarks at the same observation angle independently of the pitch-angle value. Identical SIFT feature vectors at the same yaw-angle but at different pitch-angles need not to be discriminated since we only exploit the yaw-angle.

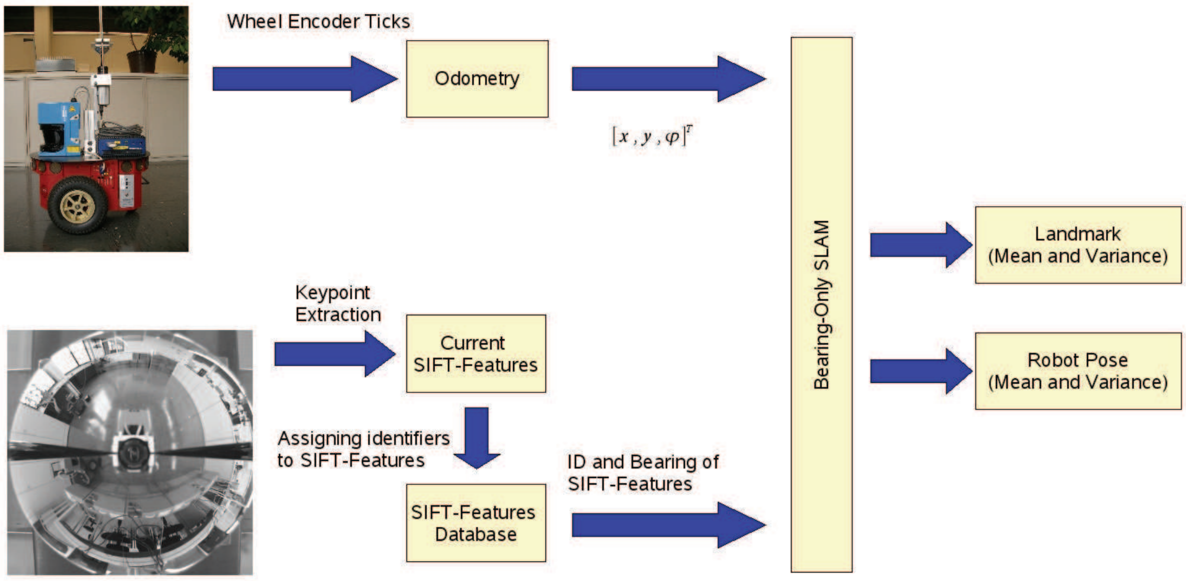

Wheel Encoder Ticks
Odometry
$[x, y, \varphi]^T$
Bearing-Only SLAM
Landmark (Mean and Variance)
Robot Pose (Mean and Variance)
Keypoint Extraction
Current SIFT-Features
Assigning identifiers to SIFT-Features
SIFT-Features Database
ID and Bearing of SIFT-Features

Fig. 9. The overall bearing-only SLAM system based on SIFT features.

6.3 Processing of an image

Each omnicam image is reduced to a 480x480 resolution. SIFT features are extracted based on the standard attributes (gaussian filter, contrast, curvature ratio). Since the omnicam image also comprises the robot and mountings of the camera, we again remove all landmarks in those areas by a simple masking operation.

6.4 Assigning identifiers to SIFT-features

The decision tree behind the identifier assignment procedure is illustrated in figure 10. The SIFT feature descriptors of the current image are compared with all the SIFT feature descriptors of the previous images. However, we only consider those images where the euclidean distance to the image acquisition pose is less than two times the maximum viewing distance of the omnicam (in our case 15m). This preselection significantly reduces the computational load of the comparisons of the descriptors. The viewing range is motivated by the typical size of free space in indoor environments.

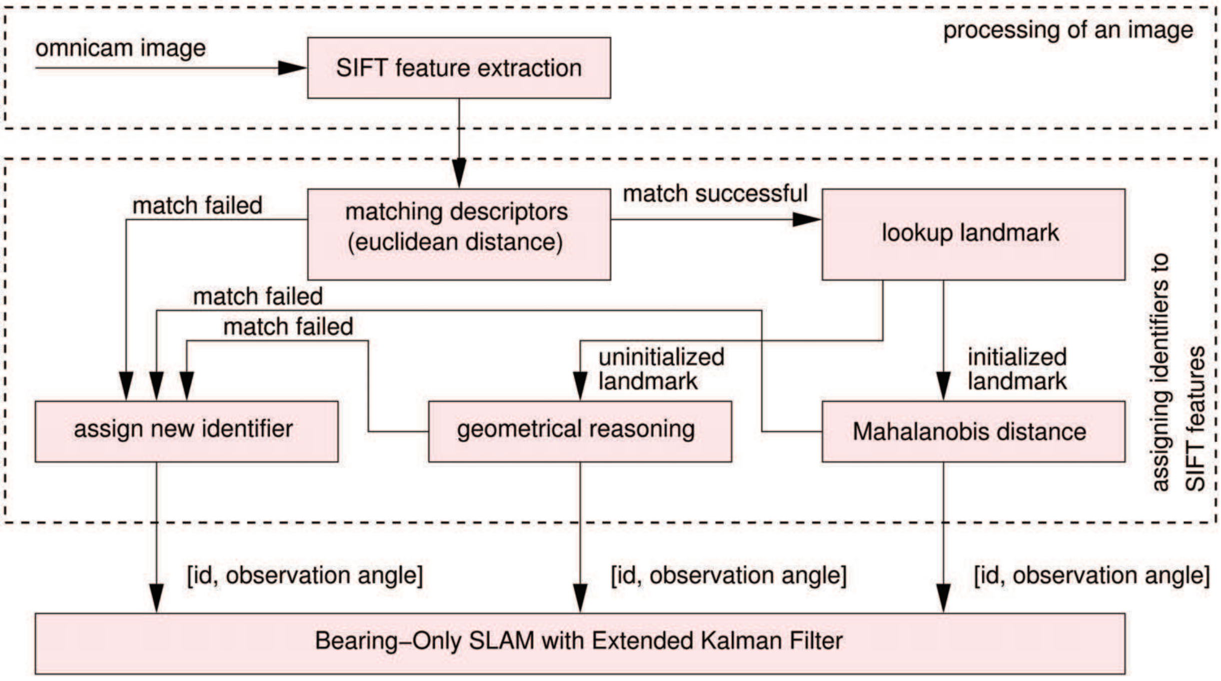


Fig. 10. The decision tree behind the identifier assignment procedure.

Next the euclidean distance between the current SIFT feature vectors and the remaining ones of the previous steps are calculated. A SIFT feature of the current image is considered as not matching an already known landmark (either initialized or not initialized landmark) if the ratio of the smallest and second smallest distance value is above a given threshold (value 0.6, see (Lowe, 2004)). In that case, this SIFT feature gets a new and unique identifier. This SIFT feature is the first observation of a potentially new landmark (first measurement of an uninitialized landmark).

Otherwise, the SIFT feature is considered as matching an already known landmark. In this case, we have to distinguish whether the SIFT feature matched an initialized or an uninitialized landmark.

In the first case, the considered SIFT feature is just a reobservation of an already known landmark which is validated by a test based on the Mahalanobis distance (Hesch & Trawny, 2005). In case of passing this test, the measurement is forwarded to the EKF as reobservation of the initialized landmark. Otherwise, the current measurement and its SIFT feature is the first observation of a potentially new landmark (first measurement of an uninitialized landmark).

In the second case, we solely have several observations (bearing-only measurements) of the same SIFT feature (uninitialized landmark) from different observation poses. Since in that case we cannot apply the Mahalanobis distance, we use geometrical reasoning for validating the reobservation. The new observation can belong to the uninitialized landmark only if its viewing direction intersects the visual cone given by the previous measurements of this uninitialized landmark. In that case, this SIFT feature is considered as a new observation of this not yet initialized landmark. Otherwise, this SIFT feature is the first observation of a potentially new landmark (first measurement of an uninitialized landmark).

6.5 Geometrical reasoning

In case of an uninitialized landmark, covariances are not yet available. Thus, we cannot apply the Mahalanobis distance to validate the assignment. Therefore, we apply a simple geometrical validation scheme that reliably sorts out impossible matches. In figure 11, *P2* denotes the current robot pose with *a* being the vector towards the previous robot pose *P1* and *c2* limiting the viewing range. At *P1* a landmark *L* has been seen with heading *b* and a maximum distance as indicated by *c1*. Thus, *L* can only be seen from *P2* in case its observation angle is in the viewing angle *r*. However, the closer the landmark is to the half-line *P1P2*, the less selective is the viewing angle *r*. In worst case, the full range of 180 degree remains. The closer the landmark is to the half-line *P2P1*, the more selective is this approach. In best case, a viewing angle close to zero remains.

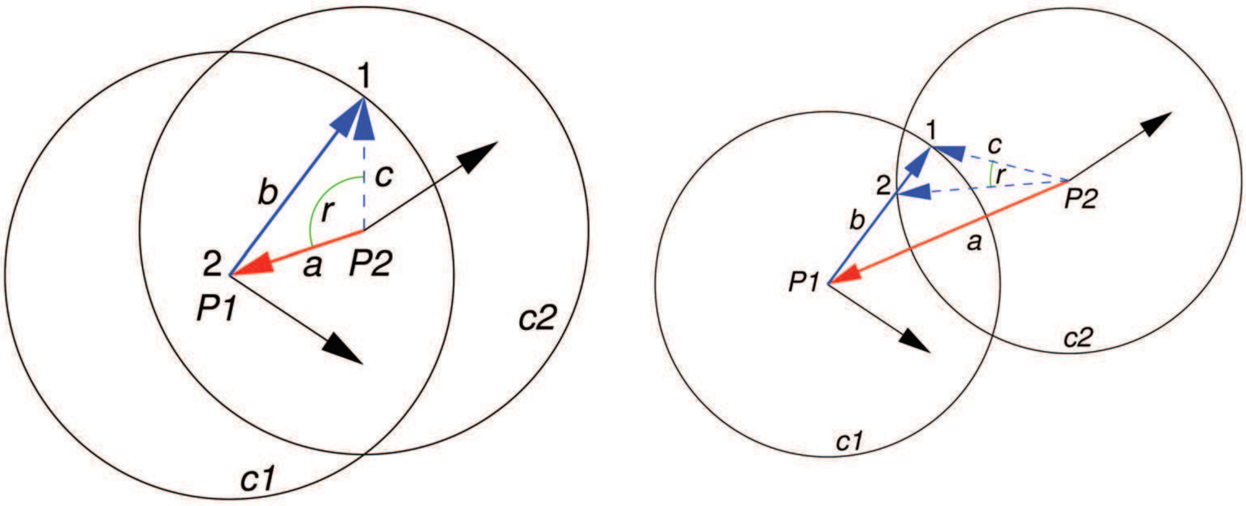


Fig. 11. Geometrical validation of matches.

6.6 Experimental setup

Due to extended experiments with our Pioneer-3DX platforms, we could meanwhile further improve the parameters of our motion model. The updated values are $\lambda_d = (0,03m)^2 / 1m$ (distance error), $\lambda_\alpha = (4\deg)^2 / 360\deg$ (rotational error) and still no drift error. The sensor model uses $\sigma_\theta^2 = (0.5\deg)^2$ as angular error of the landmark detection independent of the image coordinates of the landmark. The improved value results from the sub-pixel resolution of the SIFT feature keypoint location. The threshold of the distance metric to decide on the initial integration of a landmark is now reduced to 3. The reduced value is stricter with respect to landmark initializations. This adjustment is possible due to the higher accuracy of the angular measurements.

6.7 Experimental results

The experiment is performed in the same environment as the previous experiment but now without any artificial landmarks. The only difference is another wall that separated the free space into two sections and restricted the view. Thus, the scenario required another loop closure.

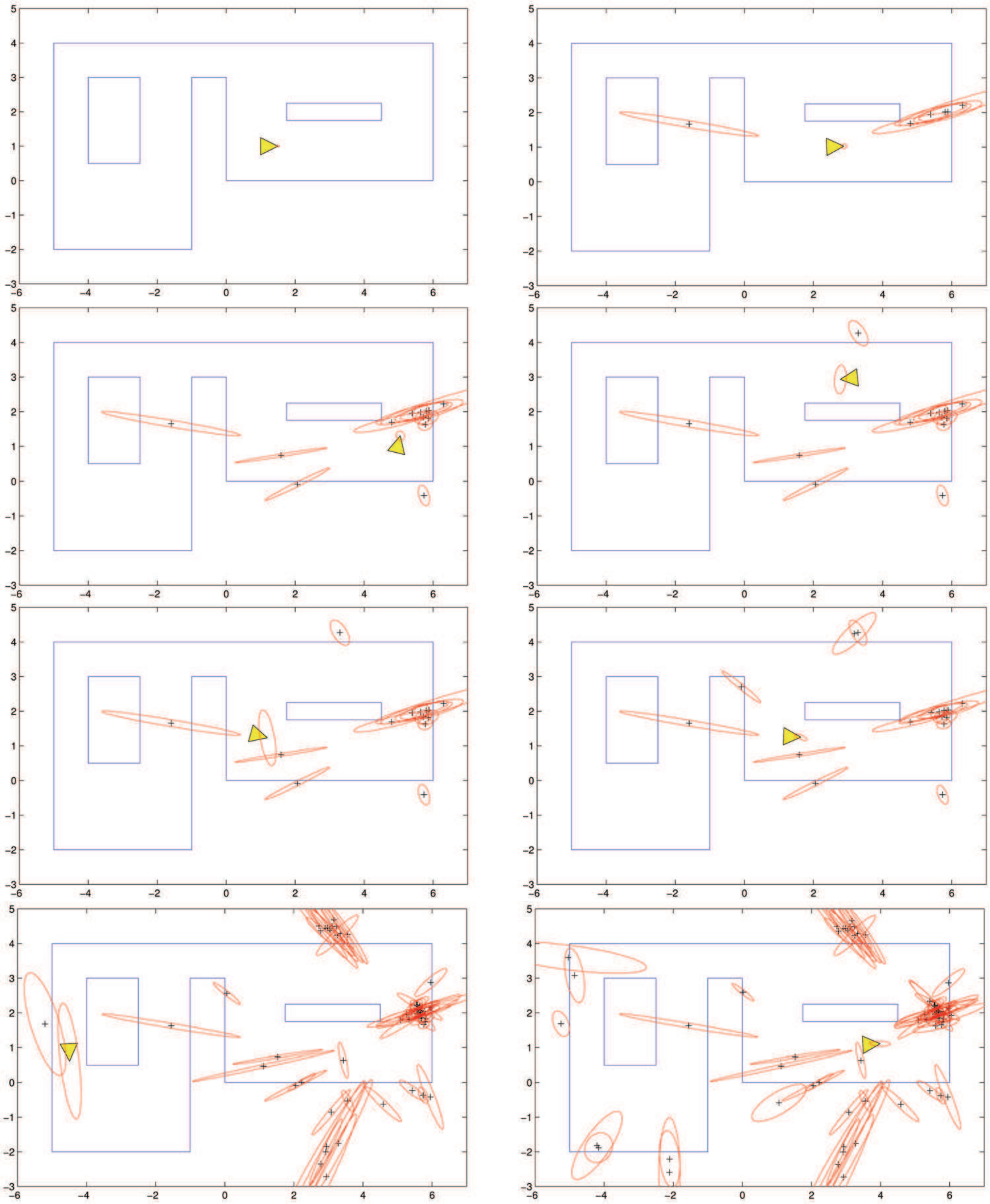


Fig. 12. The sensing steps 2, 5, 10, 18, 25, 26, 62 and 91 of a 95 step run with closing two loops using the geometrical reasoning approach with SIFT features.

Figure 12 shows the sensing steps 2, 5, 10, 18, 25, 26, 62 and 91 of a 95 step run with closing two loops. The first five images 2, 5, 10, 18 and 25 show landmark initializations and a growing robot pose uncertainty. The subsequent images 25 and 26 show the loop closure. Image 62 shows further explorations with growing uncertainty and image 95 shows the final map after another closure with reduced uncertainties. At the end of this experiment, the 2-sigma values of the robot pose are $(0{,}28m \quad 0{,}1m \quad 0{,}06rad)$. Of course, the landmark variances are still much higher and require further observations.

The experiments prove that SIFT features can be used as natural landmarks in an indoor SLAM setting. The distinctive feature of this approach is that we only use 2D landmark poses instead of 3D poses. Thus, no methods to correct image distortion or perspective are needed. The various parameters are robust and can be set in wide ranges. Thus, they can be determined with low effort.

7. Bearing-only SLAM with SIFT feature patterns

The geometrical approach is not able to restrict the viewing angle in all cases. The selectiveness depends on the positioning of the landmark relative to its observation poses. In some situations the reduced selectiveness provably led to false landmark initializations. Thus, a different approach is introduced that does not anymore depend on the geometrical configuration of the landmark and observation poses.

7.1 SIFT feature patterns

This approach improves the robustness of the re-recognition of a SIFT feature by exploiting further SIFT features in its local neighbourhood. The first extension affects the feature extraction. For each SIFT feature, the *n* nearest SIFT features (in terms of Manhattan distance in image coordinates) are determined. Now, each SIFT feature is enriched by its *n* nearest neighbours.

The second modification is the replacement of the *geometrical reasoning* box (see figure 10). The task of this box is to validate the re-recognition of an uninitialized landmark. The re-recognition hypothesis is provided by the *matching descriptor* box. The validation now compares two SIFT features by including their neighbours. For each member of the set of neighbours of the first SIFT feature, a matching entry in the set of neighbours of the second SIFT feature is searched. For efficiency reasons, this is done by calculating the correlation coefficient. The match between both SIFT features is successful as soon as at least *m* neighbouring features show a correlation coefficient value greater than a threshold t_{cc}.

This approach improves the robustness of the re-recognition since further characteristics of the local neighbourhood of a SIFT feature are considered. However, we do not exploit any geometric relationship between the neighbouring SIFT features. Thus, the SIFT feature pattern does not form a certain texture and is thus largely independent of the observation distance and angle. This is of particular importance since we do not un-distort the omnicam images.

7.2 Experimental setup

The experimental setup is the same as in the previous section. We set *n = 5*, *m = 2* and $t_{cc} = 0.8$. However, the experiments have been performed in different rooms of the same building. Figure 13 shows the hallway with artificial light and a lab room with a large window front. Thus, the lighting conditions vary extremely over a run.

Fig. 13. Another part of our buildings with very different lighting conditions.

7.3 Experimental results

Figure 14 shows the sensing steps 5, 15, 25, 34, 45, 60, 75 and 92 of a 96 step run with closing a loop. The first seven images show landmark initializations and a growing robot pose uncertainty. Image 92 shows the robot and landmark uncertainties after loop closure. At the end of this experiment, the 2-sigma values of the robot pose are $\left(0{,}14m \quad 0{,}14m \quad 0{,}04rad\right)$. Again, the landmark variances are still much higher than the robot pose uncertainties and thus require further observations.

Using neighbouring SIFT features proved to be a suitable approach to get rid of geometrical assumptions while achieving the same overall performance. Thus, this more general approach should be preferred.

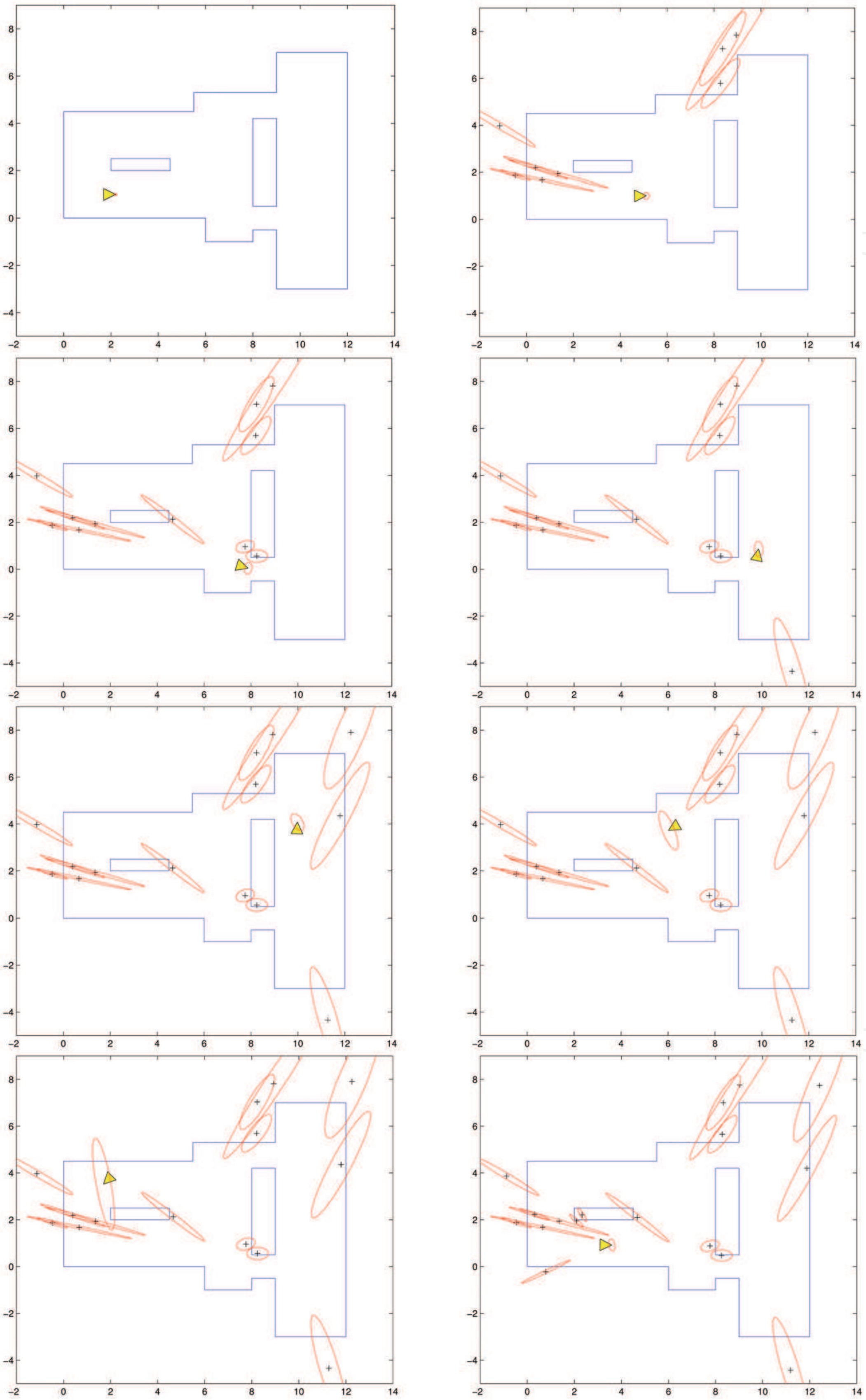


Fig. 14. The sensing steps 5, 15, 25, 34, 45, 60, 75 and 92 of a 96 step run with closing a loop. These results are based on the SIFT feature patterns.

8. Conclusion

The conducted experiments on a real platform prove that EKF-based bearing-only SLAM methods can be applied to features extracted from an omnicam image. In a first step, we successfully used artificial landmarks for the principal investigation of the performance of EKF-based bearing-only SLAM. However, service robotics applications ask for approaches that do not require any modifications of the environment. The next step was to introduce SIFT features into a bearing-only SLAM framework. We kept the idea of only estimating 2D poses of landmarks. This significantly reduces the overall complexity in terms of processing power since the state space of the Kalman filter is smaller and since the observation model is much simpler compared to 3D landmark poses. In particular the latest improvement exploiting the local neighbourhood of a SIFT feature shows stable performance in everyday indoor environments without requiring any modifications of the environment. The approach performed even under largely varying lighting conditions. Thus, the proposed approach successfully addresses the aspect of suitability for daily use as mandatory in service robotics.

9. References

Bailey, T. (2003). Constrainted Initialisation for Bearing-Only SLAM, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1966-1971, Taipei, Taiwan

Bekris, K. E. et al (2006). Evaluation of algorithms for bearing-only SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1937-1943, Orlando, Florida

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, Wiley & Sons, Inc., ISBN 0-471-06259-6, USA

Davison, A. J. et al (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 29, 6, June 2007, pp. 1052-1067, ISSN 0162-8828

Fitzgibbons, T. & Nebot, E. (2002). Bearing-Only SLAM using Colour-based Feature Tracking. *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, Auckland, November, 2002

Gil, A. et al (2006). Simultaneous Localization and Mapping in Indoor Environments using SIFT Features. *Proceedings of the IASTED Conference on Visualization, Imaging and Image Processing (VIIP)*, Palma de Mallorca, Spain, August, 2006, ACTA Press, Calgary

Hesch, J. & Trawny, N. (2005). Simultaneous Localization and Mapping using an Omni-Directional Camera. *Unpublished*

Hochdorfer, S. & Schlegel, C. (2007). Bearing-Only SLAM with an Omnicam – Robust Selection of SIFT Features for Service Robots. *Proceedings the 20th Fachgespräch Autonome Mobile Systeme (AMS)*, pp. 8-14, Springer

Intel (2008). Open Source Computer Vision Library. http://www.intel.com/technology/computing/opencv/

Kwok, N. M. et al (2005). Bearing-Only SLAM Using a SPRT Based Gaussian Sum Filter. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1109-1114, Barcelona, Spain

Lemaire, T. et al (2005). A practical 3D Bearing-Only SLAM algorithm. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2449-2454, Edmonton, Canada

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2, pp. 91-110, ISSN 0920-5691

Ortega et al (2005). Delayed vs Undelayed Landmark Initialization for Bearing-Only SLAM. *ICRA 2005 Workshop W-M08: Simultaneous Localization and Mapping*, Barcelona, Spain

Rihan, J. (2005). An exploration of the SIFT operator. http://cms.brooks.ac.uk/research/visiongroup/publications/msc/an_explaration_of_the_sift_operator_text.pdf

Roumeliotis, S. & Burdick, J. (2002). Stochastic cloning: A generalized framework for processing relative state measurements. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1788-1795, Washington, DC, USA

Schlegel, C. & Hochdorfer, S. (2005). Bearing-Only SLAM with an Omnicam – An Experimental Evaluation for Service Robotics Applications. *Proceedings of the 19th Fachgespräch Autonome Mobile Systeme (AMS)*, pp. 99-106, Springer

Sola, J. et al (2005). Undelayed Initialization in Bearing-Only SLAM. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2499-2504, Edmonton, Canada

Tamimi, H. et al (2006). Loalization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT. *Robotics and Autonomous Systems*, 54, 9, pp. 758-765, ISSN 0921-8890

Thrun, S. et al (2005). *Probabilistic Robotics*. The MIT Press, ISBN 0-262-20162-3, Cambridge, Massachusetts

Wang, X. & Zhang, H. (2006). Good Image Features for Bearing-Only SLAM. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 2576-2581, Bejing, China

Advances in
Service Robotics
INTECH