Simultaneous Localization and Mapping using an Omni-Directional Camera

Joel Hesch (Student ID#: 2400999), Nikolas Trawny (Student ID#: 3430849)
{joel, trawny}@cs.umn.edu

***Abstract*— The ability of a robot to determine its position while at the same time creating a map of a previously unknown environment (also termed SLAM) is an important step towards autonomy in mobile robotics. One well-known approach consists of estimating the robot position and orientation, as well as the map comprised of the positions of landmarks in the environment, using an Extended Kalman Filter (EKF). Commonly found in the literature are implementations using sensors, such as laser scanners, that return range and bearing measurements to the landmarks. Cameras, on the other hand, which are less expensive and more robust, only yield bearing measurements. This imposes additional challenges on the SLAM algorithm, particularly for feature initialization and data association. In this paper, we propose an implementation of bearing-only SLAM in an indoor environment using an omnidirectional camera as primary sensor. We will discuss Computer Vision algorithms for image-unwrapping and line-extraction, as well as EKF-based methods for bearing-only SLAM, followed by results from extensive simulations and initial hardware experiments on a PackBot robot.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) addresses the problem of localizing a robot in an unknown environment while at the same time mapping its surroundings. In order to take into account the system and measurement noise, several probabilistic techniques for SLAM have been investigated in the past. A well-known approach consists of fusing measurements from proprioceptive (e.g. odometry) and exteroceptive sensors (e.g. GPS, camera,...) by means of an extended Kalman filter (EKF) [1]. This filter provides an estimate of the robot position and orientation (referred to as pose), as well as an estimate for landmark locations. The landmarks are usually assumed to be stationary (static-world assumption). When no exteroceptive measurements are available, the robot can localize by dead-reckoning, i. e. propagation of its motion model by means of data obtained from odometry. Dead-reckoning has the drawback of unbounded error-growth, so that the robot will get lost after a short distance traveled. However, by repeatedly observing static external objects (landmarks), the robot can obtain a much more accurate position estimate.

A common approach in SLAM is to use range and bearing measurements towards landmarks, usually obtained by means of complex and costly laser scanners. The use of inexpensive sensors in SLAM is therefore a very desirable field of study. Small CCD cameras are being investigated as cheap, readily available, and reliable alternatives to laser range finders. Contrary to the latter, however, a single camera can only provide bearing measurements to external landmarks. Simultaneous Localization and Mapping using only bearing information is also referred to as *Bearing-Only SLAM*.

In this paper we investigate using an omnidirectional camera as exteroceptive sensor, providing us with a 360-degree view of the environment. Landmarks, such as doors and corners, will produce vertical lines in the images. We will present an algorithm allowing us to extract these vertical lines and compute their bearing relative to the robot. This bearing information will be used in the update stage of an EKF-based SLAM algorithm.

Two of the main problems encountered by SLAM are feature initialization and data association. The first problem describes the process of including a newly observed landmark in the map. The latter deals with establishing a correspondence between measurements and the originating landmark. These two problems are particularly challenging in bearing-only SLAM. The lack of additional range information prohibits landmark initialization from only one measurement and complicates data association. We will discuss possible approaches to deal with these issues.

The outline of the report is as follows: After a discussion of previous work in SLAM in section II, section III describes our method of image processing with the goal of extracting vertical lines. Section IV discusses our bearing-only SLAM implementation and gives a description of landmark initialization and data association, the two main problems in SLAM. Section V shows results of software simulations used to test and validate our algorithms. We will end the paper with a conclusion and an outlook on future work.

II. Literature Review

Compared to a wealth of literature on range and bearing SLAM, bearing-only SLAM has only been discussed in relatively few recent papers. Similarly to range and bearing SLAM, bearing-only SLAM is essentially an estimation problem and can be solved using several stochastic techniques, such as Maximum Likelihood approaches, particle filters or the extended Kalman filter, as discussed, for example, by Deans [2]. One example of early work on SLAM using a single camera is that of Chenavier and Crowley [3]. They merge odometry estimates and azimuth/elevation measurements to known landmarks using a Kalman Filter. One major limitation of this work is the fact that landmark initialization and data association are assumed as being solved.

However, landmark initialization and data association are

some of the main issues inherently problematic with bearing-only SLAM.

Landmark initialization cannot always be accomplished even when a series of robot poses and measurements is available. Bailey [4] presents a method which allows the landmark initialization to be delayed until a pair of measurements for a given landmark is well conditioned and the landmark position can be estimated with sufficient accuracy. Deans [2] proposes a batch initialization technique. This approach is applicable to smoothing, but does not allow online navigation and map building. Kwok and Dissanayake [5] present a multiple hypothesis approach for initializing an EKF. Unlikely hypotheses are eliminated from the map using a sequential probability ratio test. Davison [6] solves the initialization problem by assuming a uniform prior for the depth of a landmark. Particle filtering is then employed to recursively estimate the camera pose in 3D.

Data association is the process of matching a measurement with an already initialized landmark, or matching bearing measurements to a new landmark in order to initialize it in the map. If the landmark locations are already known, Costa *et al.* accomplish data association with a Chi-square test (Mahalanobis distance) [7]. As long as a landmark is not properly initialized, the authors propose to limit the number of possible landmark candidates by applying several plausibility checks. Landmarks that were erroneously initialized are eventually eliminated from the map using a persistence metric.

In a somewhat different approach, Thompson and Zelinsky [8] use an omnidirectional camera to learn places by extracting distinctive features and recording their specific arrangement. The robot then localizes relative to these places in a Turn Back and Look - strategy. Localization is done by means of an EKF that is initialized with depth estimates from a form of structure-from-motion algorithm. However, the authors do not describe the process of associating a consistent covariance estimate during the initialization.

Our approach will combine elements from Bailey [4] for feature initialization and from Costa et al. [7] for data association.

Before discussing the SLAM implementation, however, we will present our method for extracting bearing measurements to landmarks from the panoramic images.

III. Computer Vision System

Designing a robust computer vision system for feature detection can be a daunting task. Clearly, attempting to detect all important features in all possible environments would be intractable algorithmically and computationally. Additionally, the formulation of SLAM constrains features of interest to be static with respect to the explorer. For these reasons, a target feature set has been selected which occurs with moderate frequency in indoor areas and which fulfils the requirements listed above. The features of interest include any structural components which have long vertical edges. Some examples of features which exude this characteristic are doors, windows and corners.

The camera has been fitted with an omnidirectional lens which utilizes a pair of parabolic mirrors to give a 360-degree view of its surroundings. In the original image, the features of interest appear as radial lines. Although it may be possible to use image processing techniques to extract the features of interest from the original image, it is more reliable to unwrap the image before processing. An example image is shown in Figure 1. The image processing algorithm consists of two stages. First, each image is unwrapped to a panoramic view so that features can be more reliably detected. Second, a gradient based line detection algorithm is applied, and all non-vertical lines are rejected. Each surviving line is a measurement which will be passed to the SLAM update process.

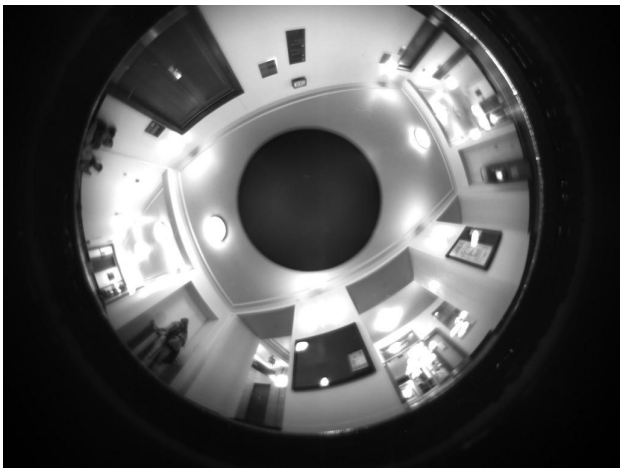

Fig. 1. Omni-Camera Image

A. Unwrapping

Before an omnidirectional image can be processed by any traditional method, it must first be unwrapped. When it comes from the camera, the image is a nonlinear mapping of a 360-degree panoramic view onto a donut. Recovering the panoramic view from the wrapped view requires the reverse mapping of points from the wrapped view onto the panoramic view. The axes of the unwrapped image are the azimuth (ϕ) and the elevation (θ). The nonlinear scaling of the elevation does not effect the orientation of vertical edges in the scene, so only the azimuth plays a role in unwrapping.

$$x_i = \rho \cos\phi + C_x \quad (1)$$

$$y_i = \rho \sin\phi + C_y \quad (2)$$

$$I_{unwrapped}(x, y) = I_{wrapped}(x_i, y_i) \quad (3)$$

Due to the nature of the nonlinear transformation there is no exact pixel-pixel match between the two images. The nearest neighbor approach has been used to accommodate any uncertainties in the reverse mapping. Although methods like bi-linear interpolation would render a more correct reverse mapping, nearest neighbor was chosen for its computational savings. In soft realtime systems it is important to cut the running time down whenever feasible. The convergence properties of SLAM are reliant on the frequency of measurement updates, so it is vital to have an algorithm which is fast as well as robust. The unwrapped image is shown in Figure 2. It

is still scaled nonlinearly with respect to the elevation, but the azimuth has been corrected. The vertical lines appear straight in the image [9].

B. Smoothing

Nearly all image processing algorithms begin by passing the image through a smoothing filter. The purpose of the smoothing filter is to remove as much pixel noise as possible so the image can be processed more reliably. The smoothing procedure recomputes each pixel value as a spacial weighted average over a patch of surrounding pixels. Mathematically this is done by convolving the image with a Gaussian kernel. The Gaussian kernel is a small square matrix whose values satisfy the well known equation for the multi-dimensional Gaussian probability density function (pdf).

$$f(x) = \frac{1}{(2\pi)^{\frac{N}{2}} det^{\frac{1}{2}}(C_x)} e^{-\frac{1}{2}(x-\mu_x)^T C^{-1}(x-\mu_x)} \quad (4)$$

$$x = [x(0), x(1), \ldots, x(N-1)] \quad (5)$$

$$\mu_x = [\mu_x(0), \mu_x(1), \ldots, \mu_x(N-1)] \quad (6)$$

$$N = 2 \quad (7)$$

The kernel k can be convolved with the image I in the following manner.

$$I_{smooth} = (I \circledast k)(x,y) = \sum_{i,j \in K} I(x-i, y-j) \cdot k(i,j) \quad (8)$$

Each pixel in the new image $I \circledast k$ takes its value from the pointwise-multiplication and summation of the surrounding original image patch, and kernel.

C. Computing the Derivative

Identification of edge pixels relies on the information contained in the image gradient. The reason is that visually detectable edges usually have a high intensity difference with respect to their surroundings. Computation of the image gradient involves convolution with an edge detector mask.

$$k_{Sobel} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (9)$$

This is a 1-dimensional Sobel edge detector mask which will give the gradient only in the x-direction. Vertical edges do not generally exhibit strong gradients in the y-direction, thus it is satisfactory to work with the 1D gradient. Another caveat to proper identification of edge pixels concerns the sign of the derivative. Some edge pixels may have large derivative values which are negative due to the direction of the gradient at that pixel. Using the magnitude of the derivative is a good alternative.

$$D_{mag} = \left| I_{smooth} \circledast k_{Sobel} \right| \quad (10)$$

In the derivative magnitude image the intensity of each pixel is a measure of how likely it is to be an edge pixel [10]. The derivative magnitude is shown in Figure 3.

D. Thresholding

Choosing the pixels in the derivative magnitude image which correspond to edge pixels is done by examining the distribution of the intensity values. Ideally, an analytic threshold could be established to separate the edge pixels from the non-edge pixels. However, due to lighting condition changes as well as other factors, the distribution of intensity values varies widely from image to image. To accommodate these variations from image to image, the threshold makes use of the first and second moments of the intensity value distribution.

$$Edge_{list} = \left\{ (i,j) \mid D_{mag}(i,j) \geq \mu_{D_{mag}} + \sigma_{D_{mag}} \right\} \quad (11)$$

E. Nonmaximum Suppression

The list of prospective edge pixels must be reduced further to produce a list which only contains the pixels right on the edges. To accomplish this, each prospective edge pixel is compared with its neighbors along the gradient direction. Only those pixels which are local maxima are chosen as true edge pixels. The direction of the gradient is restricted to be parallel to the x-axis because the gradient was only taken in the x direction. In Figure 4, the red pixels are the true edge pixels. They are shown on the derivative magnitude image to show how many potential edge pixels are rejected by thresholding and nonmaximum suppression.

F. Line Fitting

Determining which edge pixels correspond to vertical lines in the image can be accomplished in many ways. An approach with low computational complexity is to create a histogram of edge pixel counts in each column. This histogram will have high values in the bins corresponding to columns which have vertical edges. Using a statistical threshold, the peaks of this histogram can be selected.

$$Line_{list} = \left\{ i \mid Hist(i) \geq \mu_{Hist} + \sigma_{Hist} \right\} \quad (12)$$

While this algorithm is not as robust as more sophisticated approaches, it has nice computational qualities, and it strongly rejects lines which are not vertical. The result of this line fitting algorithm is seen in Figure 5.

The bearing information can be extracted directly via a linear transformation from the column pixel number. It will be fed into the update stage of the extended Kalman filter which will be presented in the next section.

IV. Bearing-Only SLAM algorithm

In the Extended Kalman Filter - framework for SLAM, the robot pose and the map are stored in form of a state vector $\mathbf{x}_k = \left[\mathbf{x}_R^T, \mathbf{x}_{L_1}^T, \ldots, \mathbf{x}_{L_n}^T\right]^T$ containing the 2D-planar robot pose at time step k, given by $\mathbf{x}_R = [x_R, y_R, \phi_R]^T$, and the landmarks denoted by $\mathbf{x}_{L_i} = [x_{L_i}, y_{L_i}]^T$. Figure 6 shows an illustration.

The Extended Kalman Filter (EKF) computes an estimate for this state vector during a prediction and an update phase. The predicted estimate is obtained through propagation of the

Fig. 2. Unwrapped Image

Fig. 3. Gradient Magnitude

Fig. 4. Nonmaximum Suppression

Fig. 5. Image with Detected Lines

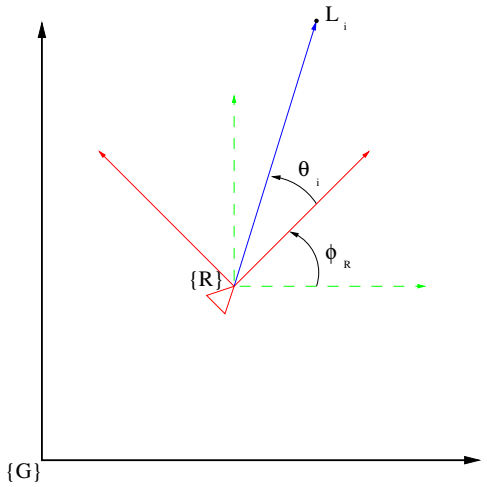

Fig. 6. Vehicle State and Measurement Model

robot's motion model. During the update phase, information from exteroceptive measurements are incorporated into the current estimate. Together with the estimate, the EKF provides a measure of uncertainty in the form of the state covariance matrix $\mathbf{P}_k$.

The motion model and the Kalman Filter equations used in the propagation and update equation will be the same as in the classical SLAM formulation with range and bearing measurements, as for example discussed in [1]. We will not discuss these equations further in this report and ask the reader to refer to the references.

The observation model for bearing-only SLAM, however, impacts the entire SLAM architecture and will subsequently be discussed in more detail.

For the bearing-only SLAM, the measurement corresponds to the true bearing θ_i from the robot to the landmark, corrupted by some measurement noise (cf. Figure 6).

$$z_i = \theta_i + n_i \tag{13}$$

This measurement is a function of the robot and the landmark state, and can be expressed by

$$\begin{aligned} z_i &= h\left(\mathbf{x}_R, \mathbf{x}_{L_i}, n_i\right) && (14)\\ &= \operatorname{atan2}\left(y_{L_i} - y_R, x_{L_i} - x_R\right) - \phi_R + n_i && (15)\end{aligned}$$

n_i is the measurement error, modeled as a zero-mean, white gaussian random variable with variance

$$E\{n_i^2\} = \sigma_\theta^2 \tag{16}$$

The EKF update equations also require the jacobian of this model, given by

$$\begin{aligned} \mathbf{H} &= \nabla_{\mathbf{x}} h(\hat{\mathbf{x}}) \\ &= \begin{bmatrix}\mathbf{H}_R & 0 & \dots & 0 & \mathbf{H}_{L_i} & 0 & \dots & 0\end{bmatrix} && (17)\end{aligned}$$

$$\begin{aligned} \mathbf{H}_R &= \begin{bmatrix}\frac{\hat{y}_{L_i} - \hat{y}_R}{\hat{d}_{LR}^2} & -\frac{\hat{x}_{L_i} - \hat{x}_R}{\hat{d}_{LR}^2} & -1\end{bmatrix} && (18)\\ \mathbf{H}_{L_i} &= \begin{bmatrix}-\frac{\hat{y}_{L_i} - \hat{y}_R}{\hat{d}_{LR}^2} & \frac{\hat{x}_{L_i} - \hat{x}_R}{\hat{d}_{LR}^2}\end{bmatrix} && (19)\end{aligned}$$

where $\mathbf{H}_R$ and $\mathbf{H}_{L_i}$ are the jacobians with respect to the robot and the landmark state, respectively. The position of $\mathbf{H}_{L_i}$ in $\mathbf{H}$ corresponds to the position of the landmark's estimate in the state vector. The quantity d_{LR} stands for the euclidian distance between robot and landmark. Hatted quantities denote estimates. From

$$z_i = \hat{z}_i + \tilde{z}_i = h\left(\hat{\mathbf{x}}_R + \tilde{\mathbf{x}}_R, \hat{\mathbf{x}}_{L_i} + \tilde{\mathbf{x}}_{L_i}\right) + n_i \tag{20}$$

and a first order Taylor expansion we can write the error between actual and expected measurement as

$$\tilde{z}_i = z_i - \hat{z}_i = \mathbf{H}\tilde{\mathbf{x}} + n_i \tag{21}$$

While, given these equations, the further implementation of the Kalman Filter is rather straightforward, the process of landmark initialization in bearing-only SLAM requires special attention.

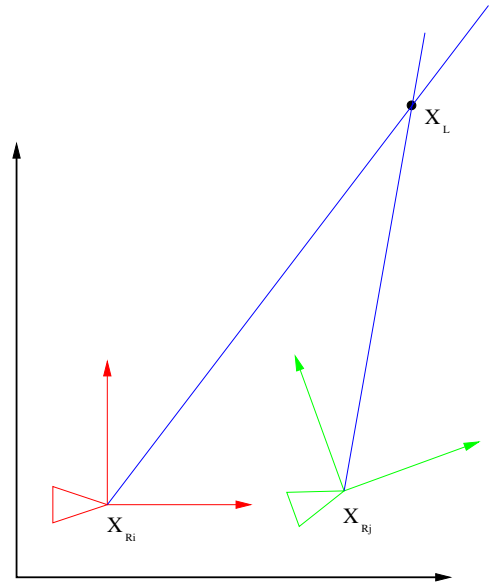

Fig. 7. Landmark Initialization

A. Bearing-Only Landmark Initialization

Contrary to the case of combined range and bearing measurements, a single bearing-only measurement does not provide sufficient information to properly initialize a landmark in SLAM. In order to uniquely determine the position of a landmark in 2D, at least two bearing measurements, z_i and z_j, from two different vehicle poses, x_{R_i} and x_{R_j}, are required.

In order to account for the cross correlations that will develop between the robot poses and the respective measurements, we will include a 'frozen copy' of the robot's pose at the time of the measurement in the state vector. This process is also referred to as 'stochastic cloning' [11]. We will use the term 'deferred measurement' for a measurement that is saved together with a state clone. Such a deferred measurement can later be used for landmark initialization.

As mentioned by Bailey [4], the exact location of the landmark in an error-free scenario can be found by computing the intersection of two lines (cf. Figure 7). The true location of the landmark would then be given by

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix}_{\text{true}} = \mathbf{g}\left(\mathbf{x}_{R_i}, \mathbf{x}_{R_j}, \theta_i, \theta_j\right) \tag{22}$$

The lines from the robot to the vehicle are given in point-slope form as

$$(y_L - y_{R_i}) = \tan(\theta_i + \phi_{R_i})(x_L - x_{R_i}) \tag{23}$$
$$(y_L - y_{R_j}) = \tan(\theta_j + \phi_{R_j})(x_L - x_{R_j}) \tag{24}$$

By equating and sorting the terms, we find the measurement function $\mathbf{g}\left(\mathbf{x}_{R_i}, \mathbf{x}_{R_j}, \theta_i, \theta_j\right)$ as

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix}_{\text{true}} = \begin{bmatrix} \frac{x_{R_i} s_i c_j - x_{R_j} s_j c_i + (y_{R_j} - y_{R_i}) c_i c_j}{s_i c_j - s_j c_i} \\ \frac{y_{R_j} s_i c_j - y_{R_i} s_j c_i + (x_{R_i} - x_{R_j}) s_i s_j}{s_i c_j - s_j c_i} \end{bmatrix} \tag{25}$$

where we abbreviate

$$s_i = \sin(\phi_{R_i} + \theta_i)$$
$$c_i = \cos(\phi_{R_i} + \theta_i)$$

In order to obtain an initial estimate of the landmark, we use this model with the state estimates and the noise-corrupted measurements, thus creating an inferred measurement whose quality depends on the error in the state estimate and our measurements. We can thus write

$$\begin{aligned} \begin{bmatrix} \hat{x}_L \\ \hat{y}_L \end{bmatrix} &= \mathbf{g}\left(\hat{\mathbf{x}}_{R_i}, \hat{\mathbf{x}}_{R_j}, z_i, z_j\right) \\ &= \mathbf{g}\left(\mathbf{x}_{R_i} - \tilde{\mathbf{x}}_{R_i}, \mathbf{x}_{R_j} - \tilde{\mathbf{x}}_{R_j}, \theta_i + n_i, \theta_j + n_j\right) \end{aligned} \tag{26}$$

Since the error of this estimate is given as

$$\begin{bmatrix} \tilde{x}_L \\ \tilde{y}_L \end{bmatrix} = \begin{bmatrix} x_L \\ y_L \end{bmatrix}_{\text{true}} - \begin{bmatrix} \hat{x}_L \\ \hat{y}_L \end{bmatrix} \tag{27}$$

we can, again using a first-order Taylor series expansion, express it in form of

$$\begin{bmatrix} \tilde{x}_L \\ \tilde{y}_L \end{bmatrix} = \mathbf{G}\tilde{\mathbf{x}} - \mathbf{W}\mathbf{n} \tag{28}$$

where

$$\begin{aligned} \mathbf{G} &= \nabla_{\mathbf{x}} \mathbf{g}(\hat{\mathbf{x}}) \\ &= \begin{bmatrix} \mathbf{G}_{R_i} & 0 & \dots & 0 & \mathbf{G}_{R_j} & 0 & \dots & 0 \end{bmatrix} \end{aligned} \tag{29}$$

$$\begin{aligned} \mathbf{G}_{R_i} &= \nabla_{\mathbf{x}_{R_i}} \mathbf{g}(\hat{\mathbf{x}}) \\ &= \frac{1}{(s_i c_j - s_j c_i)^2} \begin{bmatrix} s_i c_j (s_i c_j - s_j c_i) & -c_i c_j (s_i c_j - s_j c_i) & \dots \\ s_i s_j (s_i c_j - s_j c_i) & -s_j c_i (s_i c_j - s_j c_i) & \dots \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j c_j - (\hat{y}_{R_j} - \hat{y}_{R_i}) c_j^2 & \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j^2 - (\hat{y}_{R_j} - \hat{y}_{R_i}) s_j c_j & \end{bmatrix} \end{aligned} \tag{30}$$

$$\begin{aligned} \mathbf{G}_{R_j} &= \nabla_{\mathbf{x}_{R_j}} \mathbf{g}(\hat{\mathbf{x}}) \\ &= \frac{1}{(s_i c_j - s_j c_i)^2} \begin{bmatrix} -s_i c_j (s_i c_j - s_j c_i) & -c_i c_j (s_i c_j - s_j c_i) & \dots \\ -s_i s_j (s_i c_j - s_j c_i) & -s_j c_i (s_i c_j - s_j c_i) & \dots \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j c_j - (\hat{y}_{R_j} - \hat{y}_{R_i}) c_j^2 & \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j^2 - (\hat{y}_{R_j} - \hat{y}_{R_i}) s_j c_j & \end{bmatrix} \end{aligned} \tag{31}$$

$$\begin{aligned} \mathbf{W} &= \nabla_{\mathbf{n}} \mathbf{g}(\hat{\mathbf{x}}) \\ &= \frac{1}{(s_i c_j - s_j c_i)^2} \begin{bmatrix} (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j c_j - (\hat{y}_{R_j} - \hat{y}_{R_i}) c_j^2 & \dots \\ (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j^2 - (\hat{y}_{R_j} - \hat{y}_{R_i}) s_j c_j & \dots \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j c_j - (\hat{y}_{R_j} - \hat{y}_{R_i}) c_j^2 \\ \dots & (\hat{x}_{R_j} - \hat{x}_{R_i}) s_j^2 - (\hat{y}_{R_j} - \hat{y}_{R_i}) s_j c_j \end{bmatrix} \end{aligned} \tag{32}$$

Using the above quantities, we can express the uncertainty of the landmark's position estimate in form of the covariance matrix $\mathbf{P}_{LL}$ by

$$\mathbf{P}_{LL} = \mathbf{G}\mathbf{P}\mathbf{G}^{\mathrm{T}} + \sigma_\theta^2 \mathbf{W}\mathbf{W}^{\mathrm{T}} \tag{33}$$

The correlation with the other entries in the state vector, $\mathbf{P}_{LX}$ and $\mathbf{P}_{XL}$, by assumption of uncorrelatedness between system and measurement error, is given by

$$\mathbf{P}_{LX} = \mathbf{P}_{XL}^{\mathrm{T}} = \mathbf{H}\mathbf{P} \tag{34}$$

Obviously, the inferred measurement model will be singular if the denominator $(s_i c_j - s_j c_i)^2$ is zero or close to zero, corresponding to the case when the vectors robot-landmark at both instances of the measurement are collinear, or if an object at infinite distance were measured. The latter case could for example occur when measuring bearing to the sun or a

star, but will not be relevant for indoor applications. Near-singularity (and thus ill-conditioning of the initial landmark position estimate) also occurs when the robot only moves a very small distance between the measurements, that is, if $\mathbf{x}_{R_i}$ and $\mathbf{x}_{R_j}$ are close together. For this reason, and also to limit the computational complexity, we will defer measurements at a lower rate than the regular update frequency. In order to achieve acceptable conditioning of the initial estimate, the denominator $(s_i c_j - s_j c_i)^2$ should be bigger than a threshold when attempting the initialization. Some authors use additional criteria for determining whether a measurement pair is well-conditioned, such as a measure of how closely the resulting probability density for the landmark position estimate is approximated by a Gaussian [4]. Due to the high computational demands of this procedure we did not use it in our algorithm, although it could possibly yield more consistent estimates.

The landmark initialization procedure presented (as well as the entire EKF approach to SLAM) is based on the important assumption, that the data association problem is solved [7]. In other words, we assume we can uniquely and correctly attribute all bearing measurements to their corresponding landmark. Establishing this correspondence is actually not trivial in bearing only SLAM. In the following section we will therefore discuss a possible approach to data association.

B. Data Association

Data association is the process of matching measurements to their corresponding landmarks in the environment. In our approach, we separate this process into two sub-processes. The first deals with matching measurements to landmarks that we have already initialized into our state vector. The second is the process of matching sets of measurements to landmarks which are not in the state vector, so that we can initialize them.

1) Matching Measurements to Initialized Landmarks: In order to match measurements to landmarks which are already initialized in the state vector, we use a common gating technique that computes the Mahalanobis Distance to determine whether or not a measurement originates from a specific landmark. In the absence of noise it would be possible to discern landmarks based only on euclidian distance, in our case the difference between the measured and the expected bearing angle towards the landmark. The Mahalanobis distance, however, takes also into account the uncertainty introduced by system and measurement noise. It generates a metric of the probability that a measurement matches a specific landmark, based on the measurement z, the expected measurement $\hat{z}$, and the residual covariance S.

$$z_i = \theta_i + n_i \tag{35}$$
$$\hat{z}_i = \operatorname{atan2}(y_{L_i} - y_R, x_{L_i} - x_R) - \phi_R \tag{36}$$
$$r_i = z_i - \hat{z}_i \tag{37}$$
$$S_i = \mathbf{HPH^T} + \sigma_\theta^2 \tag{38}$$
$$m_i = r_i^T S^{-1} r_i \tag{39}$$

For Gaussian errors, the Mahalanobis Distance m_i follows the χ^2 distribution and gives a measure of confidence in the match between the estimated measurement and the real measurement. Gating of a measurement to a landmark in our state vector occurs when m_i has a value of 10.8 or less. This threshold corresponds to a match with probability of $99.8\,\%$, taken from the χ^2 distribution with one degree of freedom. Whenever a measurement is gated to a landmark already in our state vector, we do a regular update with this measurement, and then continue trying to gate the rest of the measurements in our measurement vector. It is important to note that we only gate against landmarks in the state vector whose estimates lie within sensor range. The purpose of this is to reduce the possibility of making an improper match. Since the bearing-only measurement does not provide information about the exact location of a landmark, it would actually be possible to find matches that pass the Mahalanobis distance criterion, even though the matched landmark lies completely out of range of the camera, for instance behind a wall.

2) Matching Measurements to Uninitialized Landmarks: Matching measurements to uninitialized landmarks cannot be accomplished with a gating procedure by itself. As mentioned in section IV-A we require at least two measurements to the same landmark in order to initialize it into the state vector. Without an established correspondence between each measurement and the landmarks, one possibility would be to simply initialize all possible intersections as landmark candidates. However, this approach is inefficient, because it produces large numbers of false landmarks (cf. Figure 8). To mitigate the ensuing computational complexity, we try to eliminate as many of the false landmark candidates as possible by the following strategy that involves several plausibility checks [7]. The matching of measurements to uninitialized landmarks occurs between the last three deferred measurement sets. All the intersections between the older two deferred measurements are calculated. All those which are outside of our sensor range, and those which lie on the wrong side of the robot (determined by back-projection) are discarded. Using the Mahalanobis distance metric, we try to find matches between the newest deferred measurements and the remaining plausible landmarks. Any obtained matches are then initialized into the state vector. A persistence counter is kept for each landmark in the state vector, which is incremented each time the landmark is re-observed. If we have initialized a phantom landmark (i. e. a landmark resulting from line intersections but that does not actually exist in the environment), then we should not match new measurements to it as many times as to a regular landmark. If a landmark estimate has passed out of sensor range, and does not have a sufficiently high persistence count, we will delete the landmark from the state vector.

V. Results

We tested the image processing and SLAM algorithms outlined above in extensive software simulations.

The computer vision has been implemented in C++ in GNU linux. The onboard computer has a firewire connection to which we connected a Dragonfly camera fitted with an omnidirectional lens. Images shown in this paper were taken from a test drive of the system. Due to the high computational

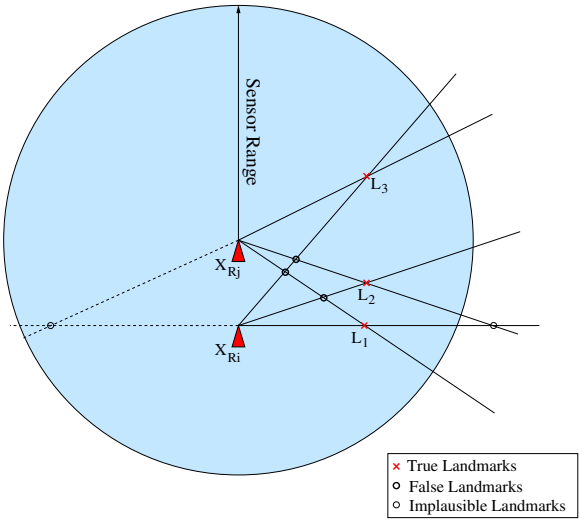

Fig. 8. Landmark candidates from possible measurement intersections. Candidates that lie outside of sensor range or to the wrong side of the robot are deemed implausible and discarded.

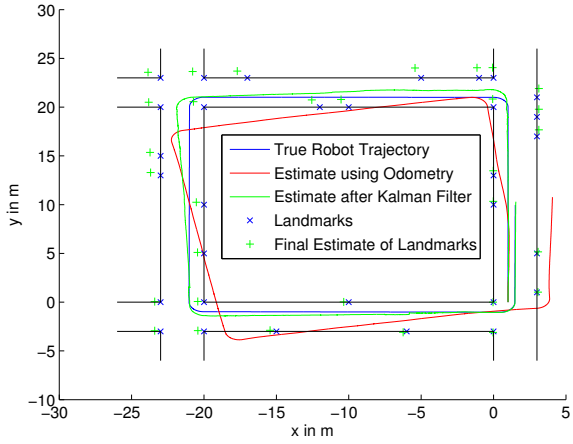

Fig. 9. Comparison of trajectories obtained through ground truth (blue), odometry (red) and SLAM (green). Note that, although the absolute position estimates of the landmarks are erroneous, the relative map is very precise.

complexity of the nonlinear mapping during the unwrapping stage, the imaging algorithm was only able to work at 3Hz on the PackBot, however, higher rates have been achieved with a more powerful desktop PC. Use of the adaptive threshold was crucial to the success of the vision system during indoor operation due to lighting changes which occurred between different parts of the environment. The system is able to robustly detect vertical lines in an image with few false positives. The reader is encouraged to see this algorithm at work in the video [12]. As the video shows, the false positives generally occur in the blurry images. These images are taken while the robot is turning quickly. Fortunately, the SLAM algorithm has a good hypothesis testing method which rejects false positives if they are not persistent. The false positive rate could be reduced even further through the use of more sophisticated edge detection techniques, however these would come at a considerable computational cost.

The results of the bearing-only SLAM simulations show that even in an office environment with relatively sparse landmark density, the algorithm achieves reasonable localization performance and, under favorable conditions, loop closure. As expected, due to the more limited measurement information, its performance is inferior compared to the regular range and bearing SLAM. Still the benefits of using exteroceptive measurements as compared to odometry only are striking (cf. Figure 9). The simulation also demonstrates the robustness of the data association strategy. All landmarks are successfully initialized and no erroneous phantom landmarks remain in the map.

The simulations further show that bearing-only SLAM is highly non-linear, resulting in nearly over-confident error estimates and bringing the EKF to the limits of stability (cf. Figure 10). However, the algorithm achieves successful loop-closure around timestep 8000, which is reflected in the corresponding notable reduction of uncertainty.

Figure 11 shows an example of the development of error for the landmark estimates. Once the landmark is initialized,

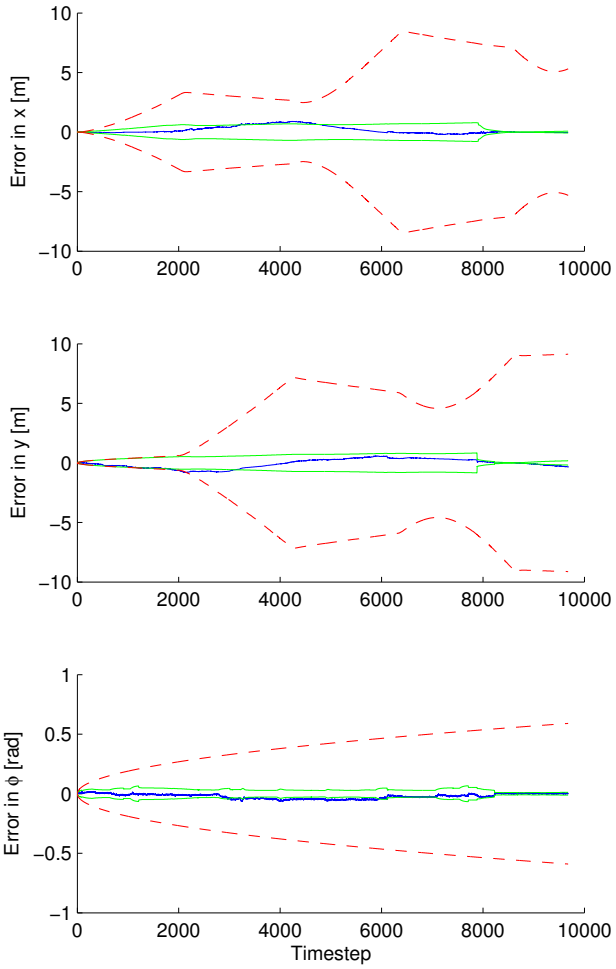

Fig. 10. Development of the robot pose error over time. Shown are only the errors for the Kalman filter output (blue), together with their 3σ bounds (green), and for comparison also the 3σ bounds for the odometry-only estimate.

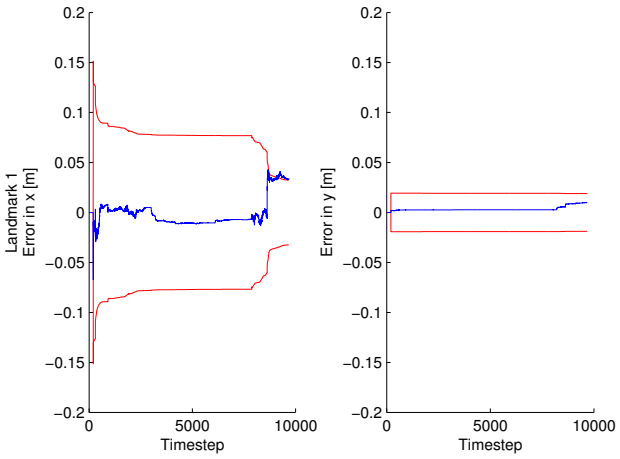

Fig. 11. Exemplary development of the error of the landmarks estimates, together with the corresponding 3σ bounds. Shown is the error for landmark 1.

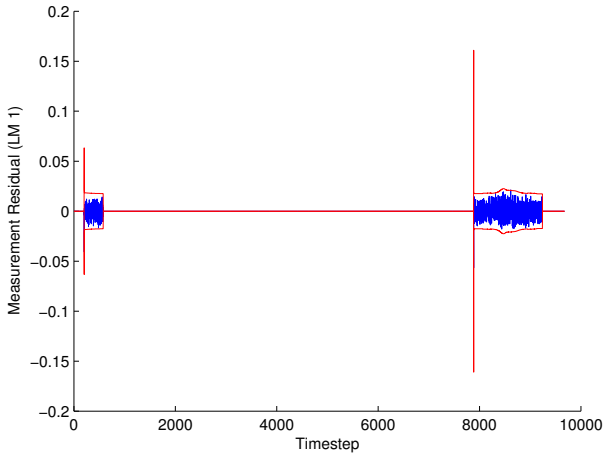

Fig. 12. The residual, i. e. the difference between true and expected measurement, for landmark 1. Note how the landmark drops out of sensor range and is reobserved shortly before timestep 8000.

the error consistently shrinks as the number of reobservations grows. As in the robot's pose error, an improvement occurs upon loop closure. However, also for the landmarks the covariance estimate tends to become overly optimistic.

Figure 12, finally, shows an exemplary realization of the residual, i. e. the difference between true and expected measurement, together with its 3σ bounds. The residual has only values different from zero when the landmark is observed, and then lies well within the confidence interval.

From the preceding discussion we can conclude that bearing only SLAM within an EKF framework is very demanding, and incurs the danger of producing erroneous, that is, overly optimistic estimates. Among the reasons for this behavior are the problem's highly non-linear structure and the little information provided by the measurements (as compared to range and bearing SLAM). As in range and bearing SLAM, we believe that the performance could be considerably enhanced by including absolute orientation information, such as from a (structural) compass. This has not yet been incorporated in the

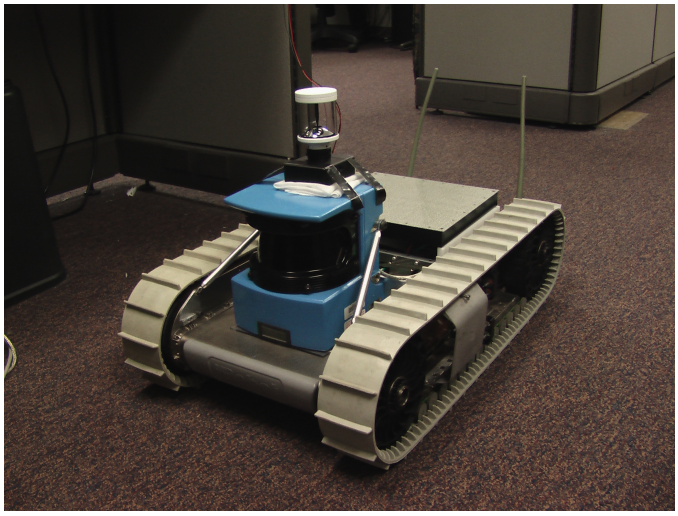

Fig. 13. The PackBot robot, equipped with a Sick Laser Scanner and an omnidirectional camera.

simulation model, but will be used in hardware experiments. The structural compass works by extracting the robot's relative orientation with respect to the hallways from laser range-finder data and then exploiting the assumed rectangular building structure to infer absolute orientation. Using this additional information will limit the linearization error and should make the algorithm more robust. We successfully implemented a controller, the structural compass, the propagation and the vision algorithm in C++ on a PackBot robot (cf. Figure 13). Due to time limitations, we were not able to fully implement the update and data association step, which will be done in the coming weeks.

VI. Conclusion

In the preceding report we have presented an approach to bearing-only SLAM using an omnidirectional camera. Our approach is specifically geared towards deployment in an indoor environment.

An algorithm for processing image as exteroceptive measurements of the environment has been presented. The purpose of the image processing is to extract bearing measurements to fixed landmarks in the environment which can be incorporated into a stochastic model of the environment which the robot can use for navigation and localization. The algorithm consists of image unwrapping, smoothing, edge detection, and vertical line identification. In general as the robustness of an algorithm increases, so does its computational complexity. As this vision system needed to work in realtime, some simplifications were made to the algorithm.

The bearing information obtained from the images was then exploited in the update phase of an extended Kalman filter-based SLAM algorithm. We presented solutions to the specific problems inherent to bearing-only SLAM, such as landmark initialization and data association.

Our algorithms were extensively tested and validated in simulation, of which we presented exemplary results.

We are currently working on completing an implementation on a PackBot robot to test the algorithms in a real-world

environment. This will confront us with additional challenges such as variable lighting conditions, calibration and poorly known system noise parameters. However, first experimental data supports our belief that our algorithms will prove efficient and robust enough to successfully navigate and localize the robot even under non-ideal conditions.

REFERENCES

[1] S. Roumeliotis, "Sensing and estimation in robotics," Department of Computer Science & Engineering, University of Minnesota," Lecture Notes, Jan. 2004.

[2] M. Deans, "Bearings-only localization and mapping," Ph.D. dissertation, School of Computer Science. Carnegie Mellon University, 2002. [Online]. Available: http://www.cs.cmu.edu/~deano/research/documents/thesis-draft.ps.gz

[3] F. Chenavier and J. Crowley, "Position estimation for a mobile robot using vision and odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, May 1992, pp. 2588–2593.

[4] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, Sept. 2003, pp. 1966–1971. [Online]. Available: http://www.acfr.usyd.edu.au/publications/downloads/2003/Bailey206/bearing_only_constrained.pdf

[5] N. Kwok and G. Dissanayake, "An efficient multiple hypothesis filter for bearing-only SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, vol. 1, Oct. 2004, pp. 736–741.

[6] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Ninth IEEE International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1403–1410.

[7] A. Costa, G. Kantor, and H. Choset, "Bearing-only landmark initialization with unknown data association," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, May 2004, pp. 1764–1770. [Online]. Available: http://voronoi.sbp.ri.cmu.edu/papers/286_WP-13_2.pdf

[8] S. Thompson and A. Zelinsky, "Accurate local positioning using visual landmarks from a panoramic sensor," in *IEEE International Conference on Robotics and Automation (ICRA)*, Washington D.C., May 2002, pp. 2656–2661.

[9] D. Burschka and G. D. Hager, "Omnidirectional vision: theory and algorithms," *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, pp. 89–96, Sept 2000.

[10] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 1st ed. Prentice Hall, 2002.

[11] S. Roumeliotis and J. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *IEEE International Conference on Robotics and Automation (ICRA)*, Washington D.C., May 2002, pp. 1788–95.

[12] J. Hesch and N. Trawny, "Video footage of a test run of the omnidirectional slam algorithm on the packbot," May 2005. [Online]. Available: http://www-users.cs.umn.edu/~joel/movie.avi