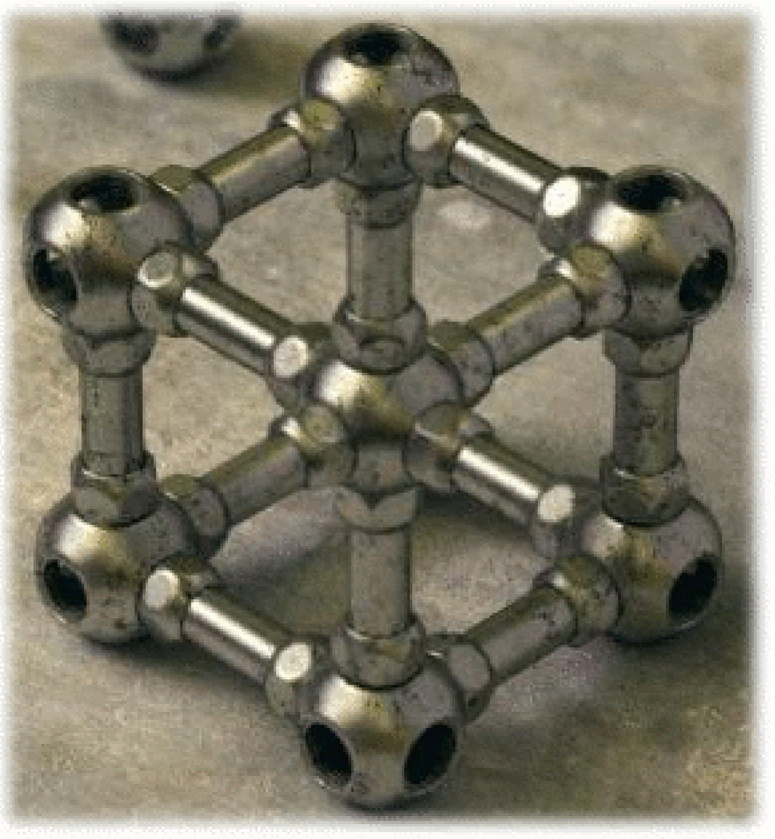

Building Object-Based Maps from Visual Input [1]

Author(s):

Antonio Henrique Pinto Selvatici
Anna Helena Reali Costa
Frank Dellaert

[1] This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

BUILDING OBJECT-BASED MAPS FROM VISUAL INPUT

ANTONIO H. P. SELVATICI[*], ANNA H. R. COSTA[*], FRANK DELLAERT[†]

[*] *LTI—Laboratório de Técnicas Inteligentes, Escola Politécnica da USP, São Paulo–SP, Brazil*

[†] *College of Computing, Georgia Institute of Technology, Atlanta–GA, USA*

Emails: antoniohps@gmail.com, anna.reali@poli.usp.br, dellaert@cc.gatech.edu

Resumo— Objects are rich information sources about the environment. A 3D model of the objects, together with their semantic labels, can be used for robot localization as well as a basis for human-robot interaction. However, traditional mapping frameworks usually build feature-based or line-based maps, without providing objects representation. On the other side, some proposed approaches for mapping objects in the environment mainly focus on adding object representations to a metric map, built using traditional SLAM techniques, blindly relying on some computer vision object recognition system. In this work we propose a framework for obtaining an *objects-based map* of the environment, together with the robot trajectory, using data acquired by an imperfect object detection and segmentation technique. The key idea is to incorporate the detected objects into a global inference framework in order to build labeled simplified geometric models for them. In the case that the detected object identities are unknown, a probabilistic data association approach is proposed, which generates maps containing objects that have an associated probability of actually existing in the environment. We found that when the objects in the resulting map have high existence probability, their locations and sizes are fully compatible with the corresponding real-world objects.

1 Introduction

Maps that represent geometric and semantic information of the objects in the environment are useful either for human visualization or as a map for robot localization and path or task planning. When an automatic system interact with human beings and communicate with them by displaying entities in the world, as may happen in Augmented Reality (AR) applications or in interactions of service robots with their users, its world representation should share symbols with that of humans. Objects are very common entities sharing the space with people, and there is evidence that people themselves use objects to represent indoors spaces (Vasudevan et al., 2007). Furthermore, if a robot is able to detect the objects in the images captured by a video camera attached to it, they are suitable features for robot localization.

In this work, we introduce *objects-based maps*, which consist of simple 3D models of certain objects in the environment, together with semantic labels indicating their class. We propose an off-line *object-based SLAM* algorithm, which uses an image sequence captured by a camera placed on a mobile robot to build these maps as well as recovering the robot's trajectory. This is done by integrating the output of an object detection algorithm into a SLAM inference engine, generating a map with objects belonging to a predetermined class or category. We deal with all aspects of the problem, including geometric inference, data association, and imperfect object recognition.

Our work is different from traditional object-based semantic mapping approaches in robotics, which are typically concerned with the cognitive environment modeling problem (Vasudevan et al., 2007; Galindo et al., 2005). The geometric aspect of the objects modeling is simplified to informing positions in a predetermined reference frame, discarding the information about object sizes and other visual information, which is not used in the mapping process. Our work also differs from traditional visual SLAM approaches, which map only some interest points in the environment (Davison et al., 2007), without attempting to provide a representation in terms of objects. In contrast, our goal is to build light-weight 3D maps of objects in the environment, together with a semantic labeling indicating the object classes (e. g., clock, TV set, table, etc.), provided by computer vision. Robot localization can takes advantage from such a representation, since maps based on objects are inherently sparser than maps based on more elementary geometric features like points or lines, allowing less expensive data association.

The remainder of this paper is organized as follows. In Section 2, we present the framework for obtaining the geometric object models and the robot trajectory from visual input data if object detection is perfect and object identities are available. Section 3 brings a the proposed framework for building object-based maps in case of unknown object identities and imperfect object detection, provided that samples from the data association space are available. In Section 4 we propose an MCMC technique to draw samples from data association space. Experimental results are discussed in Section 5, and the conclusions of this work are presented in Section 6.

2 Object-based SLAM with known data association

In this section, we show how to solve the geometric inference problem assuming the case of perfect object detection and recognition. Consider the scenario where a mobile robot explores an environment, performing a trajectory X, and captures images using a video camera attached to it. Then, the captured images are processed by object detection algorithms, specialized in detecting and segmenting instances of certain object classes. If these computer vision algorithms are also able to flawlessly recognize the objects, the data association is considered solved since each object detection is assigned to a given object in the real world.

Our objective is to perform Maximum a Posteriori (MAP) inference of the robot trajectory X and the map M using measurement data Z provided by object detection, odometer readings of the robot movement $V = \{v_i\}_{i=1}^{T}$, and the object identities J provided by object recognition. The MAP estimate is defined as

$$(X^*, M^*) = \arg\max_{(X,M)} P(X, M|V, J, Z) \tag{1}$$

where $X = \{x_i\}_{i=0}^{T}$ is the sequence of poses and M is the map. The number of objects in the map, $N(J)$, depends on the different object identities J detected by object recognition. Thus, let us define $M \triangleq \{o_j\}_{j=1}^{N(J)}$, where each object $o_j \triangleq (l_j, g_j, c_j)$ is described by an object location l_j, the geometry g_j, and the class label c_j.

The measurements $Z = \{z_{ik}, i = 1 \ldots T, k = 1 \ldots K_i\}$ provided by the object detection system are assumed to comprise the apparent contour and position of the objects detected in the image sequence captured by the robot, in addition to the detected object class. The measurement z_{ik} is the data acquired in the k^{th} object detection when the robot was in pose x_i. Hence, we assume we always have $z_{ik} = (u_{ik}, s_{ik}, a_{ik})$, where each measurement z_{ik} provides a 2D location u_{ik}, the respective apparent shape s_{ik}, and the detected class a_{ik}.

We also define the data association as $J : \{i \times \{1, 2, \ldots K_i\}, i = 0, \ldots, T\} \to \mathbb{N}+$, which is a mapping from image indices i and measurement indices, $k \in \{1, 2, \ldots K_i\}$ to object indices $j \in \{1, 2, \ldots N(J)\}$, such that $o_{J(i,k)}$ is the object detected in the image acquired at pose x_i giving rise to the measurement z_{ik}.

Assuming that we know the correct data association, we adopt a similar approach as used in traditional SLAM, except that our map includes object geometries and classes. Since object classes $C = \{c_j\}_{j=1}^{N(J)}$ are directly determined from the detected ones a_{ik} in Z and the object identities given by J, the vector of variables under inference is expressed by $\theta \triangleq (X, L, G)$, and the posterior (1) can be expressed by:

$$\begin{aligned} P(X, M|C, V, J, Z) &= P(X, L, G|V, CJ, Z) \qquad (2) \\ &\propto P(Z|\theta, C, J) P(\theta|V, C, J) \end{aligned}$$

where $P(\theta|V, C, J)$ is a prior density on trajectory and the geometric part of the map, comprising object locations $L = \{l_j\}_{j=1}^{N(J)}$ and their geometry $G = \{g_j\}_{j=1}^{N(J)}$, conditioned on object class labels C and the data association J. $P(Z|X, L, G, C, J)$ is the measurements likelihood, which does not involve odometer readings V.

The main idea we explore is that, if we roughly know the average real-world size of the objects belonging to a certain class, the apparent size of an instance in the image leads to a coarse range estimate from the robot to the object. Moreover, if we can also make assumptions about the object location, e. g., that a coach is more likely to be on the floor plane then on a table, the detected object image also gives us clues about the camera pose.

Besides the measurement model, consisting of the projection model of objects onto the image plane, we assume a prior model over object sizes depending on their classes. The prior density on the robot trajectory and the objects geometric model can be written as

$$\begin{aligned} P(\theta|V, C, J) &= P(X|V) P(L|C) P(G|C) \\ &= P(X|V) \prod_{j=1}^{N(J)} \{P(l_j|c_j) P(g_j|c_j)\} \qquad (3) \end{aligned}$$

and odometry information V casts a prior on the robot poses of the form

$$P(X|V) = P(x_0) \prod_{i=1}^{T} P(x_i|x_{i-1}, v_i) \tag{4}$$

Since no absolute localization sensor like GPS is used, the obtained map may have any reference frame. A common solution for that is defining the first pose x_0 as a constant with any value, sometimes clamping it to the origin, and making all other variables estimated with relation to it.

For our measurements likelihood, we consider that the object position in image depends on the relative displacement between the robot and object, and also on the robot orientation. The object shape is assumed independent of its position in image. Finally, we consider perfect classes detection, so that:

$$P(Z|\theta, C, J) = \prod_{i=0}^{T} \prod_{k=1}^{K_i} \{P(u_{ik}|x_i, l_{J(i,k)}) P(s_{ik}|x_i, l_{J(i,k)}, g_{J(i,k)})\} \tag{5}$$

As a result, the posterior in (1) is given by the generative model

$$\begin{aligned} P(X, M|J, Z) \propto P(x_0) \prod_{i=1}^{T} P(x_i|x_{i-1}, v_i) \prod_{j=1}^{N(J)} \{P(l_j|c_j) P(g_j|c_j)\} \\ \times \prod_{i=0}^{T} \prod_{k=1}^{K_i} \{P(u_{ik}|x_i, l_{J(i,k)}) P(s_k|x_i, l_{J(i,k)}, g_{J(i,k)})\} \end{aligned} \tag{6}$$

2.1 *Assuming Simple Geometry: Size Only*

In this work, we take g_j to be simply the object 3D dimensions, and s_k the apparent size measurements. Although the generative model of the objects shape in images can be very complex, these geometric simplifications yield more abstract object representations, that are sufficient for robot localization and task or path planning. The interesting difference with point-based monocular visual SLAM is that apparent size now yields range to objects even by a single sighting. After several sightings both object dimensions and position will be sharply determined by triangulation, obsoleting the coarse priors.

2.2 *Inference using QR decomposition*

As inference technique, we adopt the same framework as $\sqrt{\text{SAM}}$ (Dellaert, 2005). The posterior in (6) is factorized as product of Gaussian probabilities, which naturally leads (1) to be formulated as a linearized LS problem. Solving the linearized problem is part of an iterative non-linear optimization strategy, like Levenberg-Marquardt. The solution for the purely geometric SLAM problem was presented by **?**, and will be briefly shown for the sake of completeness. In this Section, we focus only the linear part.

Using linearized Gaussian models

To assure the posterior (6) is expressed as a product of Gaussian densities we define our model considering that all measurements and prior knowledge are normally distributed. Thus, the prior over objects location and size are given by

$$\begin{aligned} l_j &= \gamma(c_j) + e^l_j, \quad e^l_j \sim N(0, \Gamma(c_j)) && (7)\\ g_j &= \varsigma(c_j) + e^g_j, \quad e^g_j \sim N(0, \Sigma(c_j)) && (8) \end{aligned}$$

where e^l_j and e^g_j are the errors on the priors over objects location and size, respectively. Odometers and object measurements are also disturbed by white noise, so we can write:

$$\begin{aligned} x_i &= f(x_{i-1}, v_i) + e^x_i, \quad e^x_i \sim N(0, Q_i) && (9)\\ u_{ik} &= h^u(x_i, l_{J(i,k)}) + e^u_{ik}, \quad e^u_{ik} \sim N(0, R_{ik}) && (10)\\ s_{ik} &= h^s(x_i, l_{J(i,k)}, g_{J(i,k)}) + e^s_{ik}, \quad e^s_{ik} \sim N(0, W_{ik}) && (11) \end{aligned}$$

where e^x_i, e^u_{ik} and e^s_{ik} are, respectively, the odometry error, and the errors in the object position and size in image.

Since the functions f, h^u and h^s are, in general, non-linear, linearized versions of them are used to assure a Gaussian posterior density. Replacing the linearized version of the densities (7)-(11) in (6) yields our Gaussian posterior:

$$P(\theta|J,Z) \propto \frac{1}{\sqrt{|2\pi\mathbb{P}|}} \exp\left\{-\frac{1}{2}||A\theta - b||^2_{\mathbb{P}}\right\}, \quad (12)$$

represented in a matrix form. Each block-line in the matrix A and vector b corresponds to the coefficients of the linearized version of one of the equations (9-11), and $\mathbb{P}$ is a block-diagonal matrix with the covariances Q_i, R_{ik} and W_{ik} that weigh the summands. Maximizing the posterior (12) corresponds to finding

$$\theta^* = \arg\min_\theta ||A\theta - b||^2_{\mathbb{P}} \quad (13)$$

which is also the posterior parameters mean, with the posterior covariance expressed by $C_\theta = (A^T\mathbb{P}^{-1}A)^{-1}$.

QR factorization

The MAP inference on the posterior (12) can be transformed into an LS problem, which can be efficiently solved using QR factorization. Since (13) poses an overdetermined linear system and due to the sparseness of A, QR factorization is an efficient way to solve it. Check (Dellaert, 2005) for details. Considering $\mathbb{P}^{-\frac{1}{2}}A = Q\begin{bmatrix} R \\ 0 \end{bmatrix}$ as the QR factorization of the LS system matrix, and the constants $\begin{bmatrix} c \\ r \end{bmatrix} = Q^T\mathbb{P}^{-\frac{1}{2}}b$, the solution for the problem is given by solving the linear system $R\theta = c$, leaving $||r||^2$ as the total squared residual. If the posterior covariance is required, it can be recovered from R by doing:

$$C_\theta = (A^T\mathbb{P}^{-1}A)^{-1} = (R^TR)^{-1} = R^{-1}(R^{-1})^T \quad (14)$$

3 Probabilistic data association and mapping

In typical scenarios, object identities are not available, and thus the data association solution J must be inferred together with the geometric variables. Because J is subject to inference, the variables vector must include it. Thus, it is re-defined as $\theta \triangleq (J, \theta_J)$, where $\theta_J \triangleq (X, L_J, G_J)$ is the geometric parameters of the map and trajectory assuming the data association solution given by J. The variables we want to infer remain the same, namely the robot trajectory X and the object locations L and sizes G, which now must not depend on knowing the specific data association solution.

If all variables vectors θ_J had the same dimensionality and nature, i. e., every position in θ_J corresponded to the same physical unknown regardless of the value of J, we would estimate the unknowns by finding the expectation of the geometric variables with respect to the possible data association solutions. In computer vision, this approach is known as *correspondence-less structure-from-motion* (Dellaert et al., 2000). It can be used when the nature of the unknown vector θ_J is known *a priori*, i. e., θ_J has a fixed size and each of its components corresponds to a specific variable of the problem. The advantage of the correspondence-less structure-from-motion approach resides in taking advantage of all information that can be gathered from the data Z to infer the variables of interest, yielding optimal results even if the available data is not sufficient to certainly determine a single good data association solution. The correspondence-less structure-from-motion can be used to infer the robot trajectory X by defining the target trajectory as the expectation

$$\hat{X} \triangleq E[X|Z] = \sum_J P(J|Z) \int_{\theta_J} X\, P(\theta_J|C,J,Z) \quad (15)$$

However, when it comes to the map parameters L and G, the dimensionality of the variables vector becomes unknown, and depends on the observed data Z and the assumed object identities given by J. For instance, consider the case where data Z contains some detections of the object class "clock". If all detections are associated to a single object, the variables in θ_J are related to a single clock; on the other hand, if the "clock" detections are associated to two different objects, there are two sets of variables in θ_J related to clocks: one set describes one clock, and the other set describes the other clock.

To solve the problem of estimating the map parameters without knowing the correct number of objects in the map, this work proposes that each object have its parameters calculated separately, i. e., calculating the expectation on each individual object parameters instead of taking the expectation on the whole map at once. For such, it is necessary to develop a criterion to match the same physical object represented in variables vectors θ_J for different values of J, assigning the same *object identity* to them.

Many times, it is possible to match part of the objects in two hypotheses generated by different data associations. If two instances of J coincide that a certain group of measurements corresponds to a single object, that object is exactly the same in both vectors. Thus, we define the object identity variable $ID \triangleq \{(i,k)_1,\ldots,(i,k)_m\}$ as a set of measurement indices (i,k). We say that a data association J yields an object index ID iff: $\forall(i,k) \in ID, J(i,k) = j$ and $\forall(i,k) \notin ID, J(i,k) \neq j$, where $j \in \mathfrak{J}$ can be any object index. The expected values of the parameters related to a certain object ID are defined by the conditional expectations, which consider only the values of J where the object identity ID is found

$$\hat{l}_{ID} = \sum_{J \text{ yields } ID} P(J|ID,Z) \int_{\theta_J} l_{J(ID)} P(\theta_J|C,J,Z) \quad (16)$$

$$\hat{g}_{ID} = \sum_{J \text{ yields } ID} P(J|ID,Z) \int_{\theta_J} g_{J(ID)} P(\theta_J|C,J,Z) \quad (17)$$

where $J(ID) = j|\forall(i,k) \in ID, J(i,k) = j$ is the object index j that corresponds to the identity IDin the data association J.

3.1 Approximating the distribution on J

Enumerating the data association space is not is not practical. Since the number of possible associations grows exponentially with the number of measurements (Ranganathan, 2008), performing the summations in (15-17) exactly is not an option. However, if the distribution over data associations is approximated by a sampled version, we have $P(J|Z) \approx \frac{1}{Ns(J)} \sum_n \delta(J,J_n)$, and $P(J|ID,Z) \approx \frac{1}{Ns(ID)} \sum_n \mathbf{1}_J(ID)\delta(J,J_n)$, with $Ns(J)$ being the number of J samples, $Ns(ID)$ the number of J samples where IDoccurs and $\mathbf{1}_J(ID)$ the indicator function that indicates whether J yields . In the sampled case, we have:

$$\hat{X} \approx \frac{1}{Ns(J)} \sum_{n=1}^{Ns(J)} \hat{X}_n \quad (18)$$

$$\hat{l}_{ID} \approx \frac{1}{Ns(ID)} \sum_{n=1}^{Ns(J)} \hat{l}^n_{ID} \mathbf{1}_{J_n}(ID) \quad (19)$$

$$\hat{g}_{ID} \approx \frac{1}{Ns(ID)} \sum_{n=1}^{Ns(J)} \hat{g}^n_{ID} \mathbf{1}_{J_n}(ID) \quad (20)$$

with

$$\hat{X}_n = \int_{\theta_n} X P(\theta_n|J_n,Z) \quad (21)$$

$$\hat{l}^n_{ID} = \int_{\theta_n} l_{J_n(ID)} P(\theta_n|,J_n,Z) \quad (22)$$

$$\hat{g}^n_{ID} = \int_{\theta_n} g_{J_n(ID)} P(\theta_n|,J_n,Z) \quad (23)$$

where $\theta_n \triangleq \theta_{J_n}$. Since the density $P(\theta_n|J_n,Z)$ is assumed to be Gaussian, the expectations $\hat{X}_n$, $\hat{l}^n_{ID}$, and $\hat{g}^n_{ID}$ are simply the corresponding estimated parameters in the variables vector $\theta^*_n = \arg\max_{\theta_n} P(\theta_n|J_n,Z)$.

Algorithm 1 Building an objects-based map from data association samples J_n

```
1. Let $IDlist \leftarrow \{\}$
2. For $n$ ranging from 1 to $Ns(J)$:
   (a) Calculate the posterior $P(\theta_n|J_n,Z)$, with mean $\theta^*_n$ and covariance $C_{\theta_n}$
   (b) For $j$ ranging from 1 to $N(J_n)$
       i. Determine $ID$ so that $J_n(ID) = j$
       ii. Determine $\hat{l}^n_{ID}$ and $\hat{g}^n_{ID}$ by directly accessing these values from $\theta^*_n$
       iii. Determine $C^n_{l_j}$, the marginal covariance on $l_j$, from $C_{\theta_n}$
       iv. Let $IDlist \leftarrow IDlist \cup \{ID\}$ if $||C^n_{l_j}|| > T_s$, where $T_s$ is a spuriousness threshold
3. Find $\hat{X}$ using (18), and, for each $ID \in IDlist$, find $\hat{l}_{ID}$ and $\hat{g}_{ID}$ using (19) and (20) respectively
```

3.2 Filtering out spurious objects

The presented approach is prone to mapping more objects than the actually existing in the scenario, which we call spurious objects. The first kind of such objects are those mapped using some spurious measurements in data Z. The second kind of spurious objects occurs when a certain ID does not correspond to an actual object in the scenario.

Spurious measurements usually correspond to the detection of non consistent objects, i. e., images patches that eventually become similar to one of the objects the robot is trained to detect. Since these patches are supposed to correspond to parts of the scenario that are not detected as objects when seen from different points of view or in different time instants, few measurements are assigned to a certain spurious object of the first kind by a high probable correspondence function J. As a consequence, its marginal covariance is high, in general. To implement the elimination of spurious objects of the first kind into the framework to obtain object-based maps, we change the definition of *yielding*. Now, a certain J is said to yield IDonly if the object $o_{J(ID)} \in M_J$ is not considered as spurious, what happens if the marginal covariance of the object location $l_{J(ID)}$ has 2-norm greater than a threshold.

Furthermore, because some IDs are yielded by more samples J_n than the others, we can spot spurious objects of the second kind by assigning a probability of certain ID actually correspond to an object

$$P(ID|Z) \triangleq E[\mathbf{1}_J(ID)|Z] \approx \frac{1}{Ns(J)} \sum_{n=1}^{Ns(J)} \mathbf{1}_{J_n}(ID), \quad (24)$$

and assume as real objects all identities ID such that $P(ID|Z) > 1/2$, assuming that a certain J_n yields ID only if it is not considered as spurious of the first kind. The algorithm to build an object-based map from samples J_n is described in Algorithm 1.

Algorithm 2 The Metropolis-Hastings algorithm for sampling $P(J|Z)$

```
1. Start with a valid data association $J_0$
2. For $n$ ranging from 0 to $Ns(J)$, where $Ns(J)$ is the desired number of samples, do:
    (a) Propose a new data association $J^*$ according to an appropriate proposal distribution $q(J_n \rightarrow J^*)$
    (b) Calculate the acceptance ratio $\alpha$
    (c) With probability $\alpha$, accept $J^*$ and set $J_{n+1} \leftarrow J^*$, or $J_{n+1} \leftarrow J_n$ otherwise
    (d) Set $n \leftarrow n+1$ and return $J_{n+1}$ as a sample
```

4 Sampling the data association space

In this section, we present an MCMC-based approach to sample over the data association space. There are theoretical and practical reasons to believe that MCMC is a promising approach to perform approximate inference in the combinatorial data association space (Dellaert et al., 2000; Ranganathan, 2008). In this case, the target probability we want to sample is

$$\begin{aligned} P(J|Z) &\propto P(Z|J)P(J) \\ &\propto \int_{\theta_J} P(Z|\theta_J,C,J)P(\theta_J|C,J) \end{aligned} \qquad (25)$$

where we give the same *prior* probability to any data association. If we know how to calculate the likelihood $P(Z|J)$, a suitable way to sample the target distribution is using MCMC techniques.

In this work, we employ the Metropolis-Hastings (MH) algorithm (Hastings, 1970). MCMC methods work by simulating a Markov chain over the state space with the property of ultimately converging to the distribution of interest. Given the current chain state S_n, the MH algorithm works by accepting or rejecting a proposed new state S^* generated according to a proposal distribution $q(S_n \rightarrow S^*)$. The proposed state is accepted with probability $\alpha = \min\left(1, \frac{P(S^*)q(S^* \rightarrow S_n)}{P(S_n)q(S_n \rightarrow S^*)}\right)$, so that the chain stationary distribution becomes $P(S)$. This theoretical guarantee requires just that all proposed state transitions are reversible, i.e. $q(S_n \rightarrow S^*) > 0 \Rightarrow q(S^* \rightarrow S_n) > 0$. In addition, the MH algorithm requires $P(S)$ to be computable just up to a proportionality constant. The MH algorithm applied to sampling the data association space is described in Algorithm 2.

Despite theoretical guarantees, using MH in practice requires some extra care. Although the samples drawn from the consecutive chain states will obey $P(S)$ just when $t \rightarrow \infty$, we want this distribution to be well represented by the fewest possible samples, since computing and evaluating them demand computational effort. Well designed proposal distributions can help reducing the necessary number of samples by proposing state transitions that are more likely to be accepted, while exploring the state space. In this work, the proposal distributions were inspired on those proposed by Ranganathan (2008), and we calculate $P(J|Z)$ according to the method used by Khan et al. (2006).

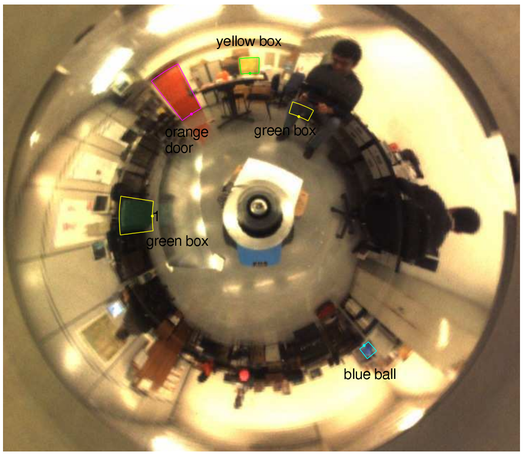

Figura 1: Example of image used in the experiment, and the detected objects. Measurements consist of slices surrounding the objects, which comprise the detected class, the position in image, the radial length, which is the projection of the object height, and the angular width. The "green box" detection at the upper-right quadrant of the image is spurious.

5 Experimental Results

We tested our approach using images captured by a robot carrying an omni-directional camera system, consisting of a video camera and a hyperbolic mirror. CMVision (Bruce et al., 2000) was used to detected colored objects placed around the environment. Although the objects we set CMVision to detect were successfully detected most of times, spurious measurements were also taken, as shown in Figure 1.

We consider objects to be well represented by cylinders, having the 3D position, diameter and height as parameters. The acquired data was used to build the objects-based map of Figure 2. Before using our object-based SLAM algorithm, the input data was filtered by associating similar measurements in consecutive images and removing those that could not be associated to any other. To generate the map, we generated 6000 data association samples, discarding the first 2500. For each sample, objects were considered spurious if their marginal covariance had the maximum singular value $\sigma_{max} > 30\text{cm}$.

To assess the quality of the obtained map, Figure 3 shows an example of projecting the obtained objects onto the acquired images using the obtained trajectory. One can see that objects with low identity probability do not actually correspond to objects in the world. On the other hand, all objects with high probability correspond to correct objects, i. e., real objects detected several times in the images, even though the projections have some shift position and size.

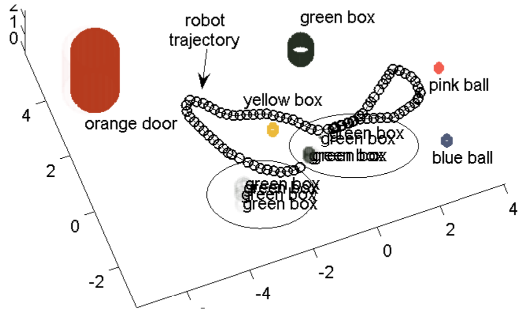

Figura 2: Objects-based map and robot trajectory, built using the proposed algorithm. The opaqueness level in objects representation indicates the identity probability. The circled objects are correctly detected as spurious.

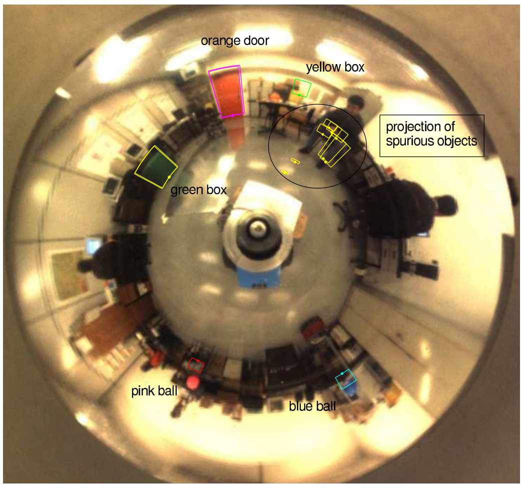

Figura 3: Projection of the mapped objects onto an image used to acquire data for the experiment. The projections of the objects with low identity probability are circled.

6 Conclusion

This work presents a novel map representation of the objects in the environment, and an offline algorithm to build it from data output by object detectors, allowing for unknown data association and spurious measurements. Results are shown for a probabilistic data association approach, whereby real objects in the map have high identity probability while objects generated by spurious detections have low existence probability.

While the current paper considered a fairly simple object detection/recognition scheme based on color segmentation, nothing in our approach prevents one from using more sophisticated object recognition methods, which is something we would like to try in future work. A drawback of our approach, however, is the high number of samples necessary to obtain good quality maps. A reliable data association technique can help lower the number of required samples considerably by pruning the data association space, and we are hopeful that methods such as JCBB (Neira e Tardos, 2001) can offer some improvement here.

Acknowledgments

Antonio H. P. Selvatici participated in this work under the grant from CNPq, Brazil, proc. no. 140.868.2005. Anna H. R. Costa is grateful to CNPq (Grant N. 475690/2008-7 and Grant N. 305512/2008-0) and FAPESP (Grant N. 2008/58737-0). The authors thank Prof. Jun Okamoto, at University of São Paulo, for having kindly lent equipment and space for data acquisition.

Referências

Bruce, J., Balch, T. e Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots, *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Davison, A., Reid, I., Molton, N. e Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM, *IEEE Trans. Pattern Anal. Machine Intell.* **29**(6): 1052–1067.

Dellaert, F. (2005). Square Root SAM: Simultaneous location and mapping via square root information smoothing, *Robotics: Science and Systems (RSS)*.

Dellaert, F., Seitz, S., Thorpe, C. e Thrun, S. (2000). Structure from motion without correspondence, *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., FernÃ¡ndez-Madrigal, J. e ez, J. G. (2005). Multi-hierarchical semantic maps for mobile robotics, *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3492–3497.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**: 97–109.

Khan, Z., Balch, T. e Dellaert, F. (2006). MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements, *IEEE Trans. Pattern Anal. Machine Intell.* **28**(12): 1960–1972.

Neira, J. e Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test, *IEEE Trans. Robot. Automat.* **17**(6): 890–897.

Ranganathan, A. (2008). *Probabilistic Topological Maps*, PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta,GA.

Vasudevan, S., Gachter, S., Berger, M. e Siegwart, R. (2007). Cognitive maps for mobile robots — an object based approach, *Journal of Robotics and Autonomous Systems* **55**(5): 359–371.