Appearance-based Concurrent Map Building and Localization using a Multi-Hypotheses Tracker

Josep M. Porta
IAS Group
Informatics Institute
University of Amsterdam
1098SJ, Amsterdam, The Netherlands
Email: porta@science.uva.nl

Ben J.A. Kröse
IAS Group
Informatics Institute
University of Amsterdam
1098SJ, Amsterdam, The Netherlands
Email: krose@science.uva.nl

***Abstract*— The main drawback of appearance-based robot localization with respect to landmark-based one is that it requires a map of the area where the robot is expected to move including images taken at known positions. In this paper, we describe a concurrent map-building and localization (CML) system developed within the appearance-base robot localization paradigm. This allows us to combine the good features of appearance-base localization such as simple sensor processing or robustness without having to deal with its inconveniences. Our localization system is based on a multi-Gaussian representation of the position of the robot. Using this kind of representation, we can deal with the global localization problem while being efficient both in memory and in execution time.**

I. Introduction

Robot localization methods can be divided in two families: methods based on landmark extraction and tracking [1]–[4], and methods based on an appearance modeling of the environment [5]–[7]. Landmark-based localization methods rely on the assumption that the position of the landmarks can be accurately extracted from the raw sensors readings. However, the transformation from sensor readings to geometric information is, in general, complex and prone to errors. As a counterpart, in the appearance-based methods the environment is not modeled geometrically, but as an appearance map that includes a collection of sensor readings obtained at known positions. The advantage of this representation is that the raw sensor readings obtained at a given moment can be directly matched with the observations stored in the appearance-based map.

A comparison between the two families of localization methods (using vision as sensory input) can be found in [8], showing that appearance-based methods are more robust to noise, certain type of occlusions and changes in illumination[1] that landmark based-methods. The main drawback of appearance-based methods is that localization is only possible in previously mapped areas. The construction of a map is a supervised process that can be quite time-consuming and that is only valid as far as no important modifications of the environment occur. While much work has been done on Concurrent Mapping and Localization (CML) using landmarks [9]–[11], this is not the case within the appearance-based approach. Recent work in this line [12] use a Kalman filter formulation to estimate the position of the robot. Using a Kalman filter only one hypothesis can be tracked at at time and, due to this, this work does not exploit all the potential of the appearance based framework as, for instance, the ability to perform global localization (i.e., the localization without any prior information on the robot's position).

In this paper, we introduce a system that is able to perform CML within the appearance-based approach. With this aim, we replace the static, pre-defined map of the environment used in appearance-based localization by an approximation to it obtained and refined by the robot as it moves in the environment. The basic idea we exploit in our system is that, if the robot re-visits an already explored area, it can use the information previously stored to reduce the uncertainty on its position. Additionally, the improvements on the robot's position can be back-propagated to map points stored in previous time slices using trajectory reconstruction techniques [13]. The result is a correction of both the robot's position and the map and, thus, we achieve the objective of concurrently localize and build a correct map of the environment.

Our approach is closely related with the work on map building on cyclic trajectories [14], [15] that is usually done using range sensors [16]. Range sensors (such as sonars or laser scanners) provide distance information (position of obstacles w.r.t. the robot) that can be integrated into a geometric map of the environment. Our sensor readings (images) do not provide geometric information and, thus, our map is not a geometric/explicit map of the environment but an implicit map in the space of features derived from images. Due to the different nature of the maps, the main issues of our approach are different form the relevant issues in existing work. For instance, when using range-based sensors, one of the main issues is that of sensor matching (i.e., to determine the translation and rotation for a given sensor reading so that the match with the map is maximal). In our case, no sensor matching is necessary (images are directly compared in the feature space) but the map representation becomes more complex (it is not just a collection of 2D points, but a set of Gaussian mixtures with associated image features).

[1]When a edge detector is used to pre-process the images.

We first describe the formalism we use in our localization framework both for robot's position and for map representation. Next, we describe how to estimate the position of the robot (assuming we have a map). After that, we describe how to extract features from the input images and how to on-line approximate the feature-based map necessary for localization. Finally, we show the results obtained with the new CML system, and we conclude summarizing our work and and extracting some conclusions out of it.

II. Multi-Gaussian Based Localization

Probabilistic systems for robot localization can be classified according to the flexibility of the mechanism used to represent the probability distribution on the robot's position. Systems based on the Kalman filter [3] represent the robot's position using a single Gaussian. These systems are effective avoiding the error on the robot's location to grow without any bound (as it would happen using only odometry for localization). Additionally, the simplicity of the model used on these systems allows for formal demonstration on the convergence of the localization (and of the associated mapping process, if any). However, using a single Gaussian, it is not possible to track more than one hypothesis at the same time and, due to this, the Kalman-based localization systems are unable to deal with the global localization problem (i.e., to determine the location of the robot without any prior knowledge on the robot's position) or with the kidnap problem. Probabilistic occupancy grids [11], [17] and particle filters [18], [19] can be used to solve the global localization problem. In the probabilistic occupancy grids framework, the area where the robot is expected to move is discretized in small cells and the system maintains the probability for the robot to be in each one of these cells. In the particle filter, the robot position is estimated using a set of discrete samples, each one with an associated weight to represent its importance. Both occupancy grids and particle filters are extremely flexible but also computationally expensive in memory and in execution time per time slice.

Multi-Gaussian probability representations [20]–[22] are an option that is between Kalman-filter systems and particle filter. These systems can track more than one hypothesis simultaneously but they do not reach the degree of flexibility of particle filters. However, for practical purposes, an approximation of the probability on the robot's position as a set of Gaussian functions is accurate enough and it has the advantage of being highly intuitive. Additionally, Multi-Gaussian systems use less resources (in memory and time) than particle filters and, of course than occupancy grids. A comparison between multi-Gaussian representations and other formalisms for robot's localization can be found in [23] showing the advantages of multi-hypotheses trackers in terms of localization accuracy and efficiency. As a drawback, working with a multi-hypothesis tracker poses the problem of data association (i.e., to determine which piece of the sensory information to use to update each Gaussian hypothesis). This data association problem is not present in particle filtering or probabilistic occupancy grids.

In this paper, we use the multi-Gaussian approach introduced by Jensfelt and Kristensen in [22]. Our contribution is that, while Jensfelt and Kristensen use a pre-defined map of the environment, our system is able to on-line build the map as the robot moves.

III. Robot Position Estimation

The probabilistic localization methods aim at improving the estimation of the pose (position and orientation) of the robot at time t, x_t, taking into account the movements of the robot $\{u_1, \ldots, u_t\}$ and the observations of the environment taken by the robot $\{y_1, \ldots, y_t\}$ up to that time[2]. Formally, we want to estimate the posterior $p(x_t|\{u_1, y_1, \ldots, u_t, y_t\})$. The Markov assumption states that this probability can be updated from the previous state probability $p(x_{t-1})$ taking into account only the last executed action u_t and the current observation y_t. Thus, we only have to estimate $p(x_t|u_t, y_t)$. Applying Bayes we have that

$$p(x_t|u_t, y_t) \propto p(y_t|x_t)\, p(x_t|u_t), \tag{1}$$

where the probability $p(x_t|u_t)$ can be computed propagating from $p(x_{t-1}|u_{t-1}, y_{t-1})$ using the action model

$$p(x_t|u_t) = \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_{t-1}, y_{t-1})\, dx_{t-1}. \tag{2}$$

Equations 1 and 2 define a recursive system to estimate the position of the robot.

We use a Gaussian mixture $X_t = \{(x^i_t, \Sigma^i_t, w^i_t) \,|\, i \in [1, N]\}$ to represent the position of the robot. Thus

$$p(x_t|u_t, y_t) \propto \sum_{i=1}^{N} w^i_t \phi(x_t|x^i_t, \Sigma^i_t),$$

with $\phi(x_t|x^i_t, \Sigma^i_t)$ a Gaussian centered at x^i_t and with covariance matrix Σ^i_t. The weight w^i_t ($0 < w^i_t \leq 1$) provides information on the certainty of the hypothesis represented by the corresponding Gaussian.

The motion of the robot is modeled as

$$x_t = f(x_{t-1}, u_t, v_t), \tag{3}$$

with v_t a Gaussian noise with zero mean and covariance Q. With this, the application of the action model (equation 2) amounts at applying equation 3 to each one of the components in X_t. Using a linear approximation,

$$\begin{aligned} x^i_{u_t} &= f(x^i_{t-1}, u_t), \\ \Sigma^i_{u_t} &= F \Sigma^i_{t-1} F^\top + G Q G^\top, \end{aligned} \tag{4}$$

with F the Jacobian of f with respect to x^i_{t-1} and G the Jacobian of f with respect to v_{t-1}. With the set $X_{u_t} = \{(x^i_{u_t}, \Sigma^i_{u_t}, w^i_t) \,|\, i \in [1, N]\}$ we define

$$p(x_t|u_t) \propto \sum_{i=1}^{N} w^i_t \phi(x_t|x^i_{u_t}, \Sigma^i_{u_t}).$$

[2]In our notation, the Markov process goes through the following sequence $x_0 \xrightarrow{u_1} (x_1, y_1) \xrightarrow{u_2} \ldots \xrightarrow{u_t} (x_t, y_t)$.

After we have $p(x_t|u_t)$, we have to integrate the information provided by the sensor readings (see equation 1). At this point, we assume we have a map of the environment from which we can define $p(y_t|x_t)$ as a set of Gaussian functions with parameters $Y_{y_t} = \{(x^j_{y_t}, \Sigma^j_{y_t}, w^j_{y_t}) \,|\, j \in [1, N']\}$. In next section, we describe how to create and update the set Y_{y_t}. From Y_{y_t} we can define

$$p(y_t|x_t) \propto \sum_{j=1}^{N'} w^j_{y_t} \phi(x_t|x^j_{y_t}, \Sigma^j_{y_t}).$$

If Y_{y_t} has no components ($N' = 0$), the estimation on the robot's position obtained applying equation 4 can not be improved and we have

$$x^i_t = x^i_{u_t},$$
$$\Sigma^i_t = \Sigma_{u_t}.$$

If $N' > 0$, we have to fuse the Gaussian functions in X_{u_t} with those in Y_{y_t}. The direct application of equation 1 amounts to associate (i.e., to find a Gaussian that represents the multiplication of) each one of the elements in X_{u_t} with those in Y_{y_t}. This would produce a quadratic ($N \times N'$) increment of the number of hypotheses. Too keep the number of hypotheses under a reasonable limit, we will only associate elements of X_{u_t} and Y_{y_t} if they are close each enough meaning that they are different approximations of the same positioning hypothesis. This arises the problem of *data association*: to determine which elements on Y_{y_t} and on X_{u_t} refer to the same hypothesis. We perform the data association using the same criterion used in [22]. For each couple (i, j) with $(x^i_{u_t}, \Sigma^i_{u_t}, w^i_t) \in X_{u_t}$ and $(x^j_{y_t}, \Sigma^j_{y_t}, w^j_{y_t}) \in Y_{y_t}$ we compute the innovation as

$$\upsilon_{i,j} = x^i_{u_t} - x^j_{y_t}$$
$$S_{i,j} = \Sigma^i_{u_t} + \Sigma^j_{y_t},$$

and we assume that hypotheses on the robot position i and sensor reading j match if the following condition holds

$$\upsilon_{i,j} S^{-1}_{i,j} \upsilon^\top_{i,j} \leq \gamma. \quad (5)$$

If there is a match, we update of hypothesis i with the sensor information j is done using the *Covariance Intersection* rule [24]. This rule permits filtering and estimation to be performed even with the presence of the unmodeled correlations in the sensor readings that are so common in real-world problems. This rule updates the covariance matrix and the average as

$$(\Sigma^i_t)^{-1} = (1-\omega)(\Sigma^i_{u_t})^{-1} + \omega(\Sigma^j_{y_t})^{-1}$$
$$x^i_t = \Sigma^i_t[(1-\omega)(\Sigma^i_{u_t})^{-1} x^i_{u_t} + \omega(\Sigma^j_{y_t})^{-1} x^j_{y_t}], \quad (6)$$

with $\omega = |\Sigma^i_{u_t}|/(|\Sigma^i_{u_t}|+|\Sigma^j_{y_t}|)$. If the information provided by the sensors is more reliable than the current one, this update results in a reduction of the uncertainty of hypothesis i (a reduction in the size of Σ^i_t w.r.t. $\Sigma^i_{u_t}$).

Hypotheses on the state not matched with any Gaussian on Y_{y_t} are just keep without any modification, but the weight update described below. Sensor components not matched with any state hypothesis have to be introduced as new hypotheses on X_t. This allow us to deal with the kidnap problem.

The confidence on each hypothesis in X_t is represented by the corresponding weight w^i_t. Following [**?**], it seems clear that, the more the sensor readings recently supporting a given hypothesis the larger the confidence. Thus, we increase w^i_t for the hypothesis that are properly matched in the *data association* process

$$w^i_{t+1} = w^i_t + \alpha(1 - w^i_t), \quad (7)$$

with α a learning rate in the range $[0, 1]$. For the not matched hypothesis, the confidence is decreased as

$$w^i_{t+1} = (1 - \alpha) w^i_t. \quad (8)$$

Hypotheses with too low weight are removed from X_t.

IV. Image Feature Extraction

Our sensory input for localization are images taken by the camera mounted on the robot. A problem with images is their high dimensionality, resulting in large storage requirements and high computational demands. To alleviate this problem, Murase and Nayar [25] proposed to compress images (z) to low-dimensional feature vectors (y) using a linear projection

$$y = W\, z.$$

In the work of Murase and Nayar, the projection matrix W is obtained by principal component analysis (PCA) on a supervised training set ($T = \{(x_i, z_i)| \; i \in [1, L]\}$) including images z_i obtained at known states x_i. However, in our case, we don't have a training set. For this reason, we use a standard linear compression technique: the *discrete cosine transform* (DCT) [26], [27]. We select this transform since, PCA on natural images approaches the discrete cosine transform in the limit. The DCT computes features from images in a linear way using a projection matrix W defined as

$$W_{j,k} = z_k \, \cos\left(\frac{\pi}{n} j(k - 1/2)\right) \quad (9)$$

with j the number of the feature to be extracted, z_k the k-th pixel of image z (considering the image as a vector) and n the total number of pixels in the image. For a given image we compute a set of d features (d is typically around 10). Those features capture a very general description of the image, omitting the low level details and making the comparison between images more meaningfully.

V. Environment Mapping

The manifold of features y can be seen as a function of the pose of the robot x, $y = g(x)$. The objective of a appearance-based mapping is to approximate g^{-1} since this gives us information about the possible positions of the robot given a set of features.

At a given time, we have a pair (X_t, y_t) with y_t a set of features and X_t the estimation of the position of the robot. We can use the sequence of such pairs obtained as the robot moves as a first approximation to the map needed for localization. Thus, our map can be initially defined as $M = \{Y_{y_t}, y_t)\}$, with $Y_{y_t} = X_t$.

As explained in section III, when the robot re-observes a given set of features y_t, we can use Y_{y_t} to improve the location of the robot after the application of the action model X_{u_t}. Due to noise a given observation is never *exactly* re-observed. So, we considerer that two observations y and y' are equivalent if $\|y-y'\| < \delta$. This discretizes the space of features with granularity δ. Thus, if y ($\pm\delta$) is re-observed, we can use an old estimation on the robot's position to improve the current one. However, we can also use the information the other way around, we can improve Y_{y_t} using the additional information provided by the current X_{u_t}. What we need to do is to introduce new information into a given Gaussian mixture and a procedure to do that is exactly what we have described in section III. Therefore, we can improve Y_{y_t} using this same procedure but here the roles are swapped: in previous section we update the position of the robot using the sensor information and now we adjust the map using the information provided by the position of the robot. So, we have to swap X_{u_t} with Y_{y_t}, x_{u_t} with x_{y_t}, and i with j. The only difference is that we assume the environment to be static and, thus Y_{y_t} is not updated by any action model.

At a given moment, the robot is at a single position. So, when the state X_{u_t} includes more than one hypothesis, we are uncertain about the robot's location and, consequently, any map update using X_{u_t} will include incorrent map modifications that we will need to undo later on. To avoid this problem, we use a conservative strategy in the map update: when X_{u_t} is not a single hypothesis (i.e., a single Gaussian) no map update is done. If X_{u_t} is a set of Gaussian functions (or a uniform distribution) and, due to the state update, it becomes a single Gaussian, we use the path followed by the robot and the backward action model to add new points to the map corresponding to the time slices where the robot was unsure about its location (and that now we now can unambiguously determine). Additionally, if X_t is a single Gaussian and the state update results in a reduction of its covariance, we also backpropagate this error reduction to previous map points. The result of these two backpropagation process is to add new points to the map (i.e., to extend the map) and to improve the estimation of previously stored map points (i.e., to improve the quality of the map).

The mapping strategy just outlined allow us to build an appearance-based map of a given environment along the paths followed by the robot. When the map M is empty, we need to now the initial position of the robot (the map is build with respect to this position), but when M has some elements, global localization is possible. To perform global localization we have to initialize the system with X_0 containing a uniform distribution. As soon as the robot observes features already stored in the map (i.e., it visits an already explored area) new hypotheses are added to X_t using the data stored in the corresponding map points. If one of these new hypotheses becomes relevant enough, we consider it as the correct position of the robot, the rest of

```
Multi Hypotheses CML(M):
  Input: M, the map.
      If M is empty X ← {(x = 0, Σ = 0, w = 1)}.
      else X ← Uniform distribution.
Do forever:
  Update X according to the action model (equations 4).
  Get an image z.
  Compute the features y of z using the DCT (equation 9).
  (y, Y) ← arg min_{∀(y',Y')∈M} ||y − y'||
  if ||y − y'|| > δ then
      M ← M ∪ {X, y}
  else
      Associate elements in X and Y (equation 5).
      Update X:
          For the associated elements use equations 6, 7.
          For the non-associated decrease weight (eq. 8).
          Remove elements in X with too low weight.
      if X is a single Gaussian (x)
          if x associated with y ∈ Y
              Update y using equations 6, 7.
          else
              Add x to Y.
          Decrease weight for y' ∈ Y, y' ≠ y (eq. 8).
          Remove elements in Y with too low weight.
```

Fig. 1. The multi hypothesis tracking, appearance-based CML algorithm.

tentative hypothesis are removed, and the new information on the robot position is backpropagated adding points to the map.

Figure 1 summarizes the CML algorithm we introduce in this paper.

VI. Experiments and Results

We tested the proposed CML system mapping a corridor 25 meters long. We drive the robot using a joystick all along the corridor and back to the origin. Figure 2-A shows the path followed by the robot according to odometric information. As we can see there is a large error in the final position of the robot (about 40 cm in the X axis, 300 cm in the Y, and 20 degrees in orientation). In the figure, each one of the arrows correspond to a robot's pose where a map point is stored.

Our CML system detects that the final point in the trajectory is close to the initial one comparing the features of the images obtained at those points. This coincidence allows a drastic reduction on the uncertainty on the robot's location and this reduction can be back-propagated improving the quality of the map. Figure 2-B show the corrected map points after the loop is closed. We can see that the correction affects mainly to the points close to the end of the trajectory, where the back-propagated information is more certainty that the previously stored one: close to the beginning of the trajectory the initial information is more reliable than the back-propagated one and the map points are not modified. In figure 2-B, the bright red arrows correspond to poses where there is perceptual aliasing: the *same* set of features are observed from all the positions plotted in red. Perceptual aliasing is one of the reasons why we need to keep track of many hypotheses simultaneously.

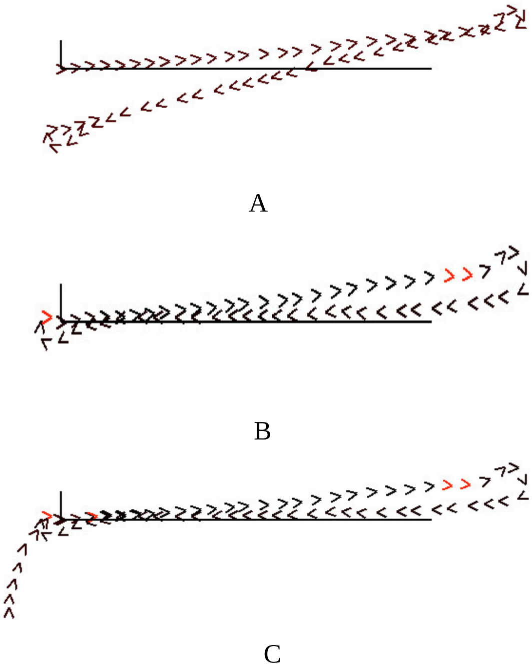

Fig. 2. Path of the robot used for the tests. A- Using only odometry, B- The map is corrected when the loop is closed C- Adding new portions the map (to do that the robot performs global localization).

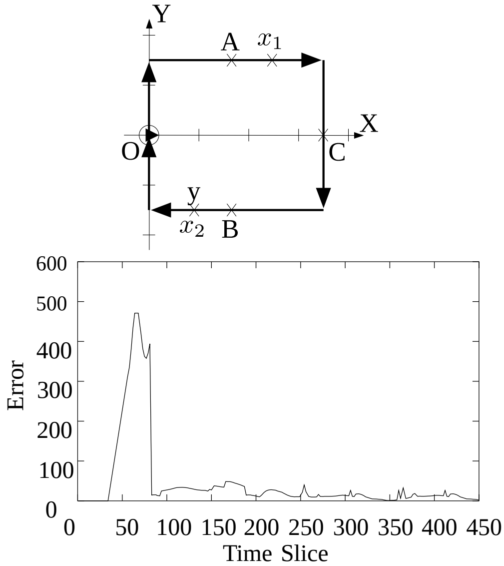

Fig. 3. Top: The circuit used for the kidnap experiment. Bottom: the error in localization for different time slices.

Another situation where multiple hypothesis should be considered is when performing global localization: when the robot is started at a unknown position (but in a previously mapped area) or after a kidnap. In the experiment reported in figure 2-C, we started the robot in a non-mapped area and we drive it to the beginning of the corridor previously mapped. In this case, the robot operates with a uniform distribution about its position. When consistent matches with already existing map points occur, the robot determines its position, the uniform distribution is replaced by a Gaussian defined from the map and new map points along the robot path are added.

In a third experiment we tested the ability of our system to deal with the kidnap problem, even while the map is in early phases of its construction. We move the robot in a close circuit of 3.5×3 meters (figure 3-top) departing from position O. At the first loop, at position A the robot is kidnapped: lifted and displaced to position B and rotated $180°$. Thus, according to odometry the robot moves from A to C while it is actually is moving from B to O. The kidnap introduces a large error while the map is in its initial construction phase (observations initially assigned to points in the path from A to C are actually occurring in the path from B to O). When the robot gets to O a new hypothesis (the correct one) is introduced into the state. This new hypothesis is reinforced by the observations and, after few time slices, it becomes the most relevant one, the wrong hypothesis is removed and, in the following iterations over the circuit, the map is corrected. We can use an example to show how the map correction works. Initially, the incorrect association $([(x_1, \Sigma_1, w_1 = 1)], y)$ is stored in the map. In the second loop, this map point is updated to $([(x_1, \Sigma_1, w_1 = 1), (x_2, \Sigma_2, w_2 = 1)], y)$. In the following iterations, the weight of the association (x_1, y) decreases sine it is not observed again while the correct association (x_2, y) is reinforced. After few loops, the association (x_1, y) becomes weak and it is eventually removed from the map.

Figure 3-Bottom shows the error in localization as the robot moves around the circuit. The large errors at time slices $40 - 90$ are caused by the initial kidnap but, as the robot gets to O again, the error is canceled and keep low. Every loop around the circuit takes 100 time slices.

VII. Discussion and Future Work

We have introduced a system that is able to simultaneously build an appearance-map of the environment and to use this map, still under construction, to improve the localization of the robot. The on-line construction and update of the map allow us to overcome the major hurdles of traditional appearance-based localization. First, the robot can operate in previously unknown areas. Second, we can deal with changes in the environment: new observations obtained at already explored positions are added to the map, the old observations at those position are not used any more and they are slowly forgotten. Finally, the way in which the map is built guarantees a uniform sampling of the feature space and not of the geometric space, as it happens in normal appearance-based localization. Sampling uniformly the feature space is essential for achieving a good localization since the sensor model is based on the similarities (i.e., the distances) in that space.

An implicit assumption in our mapping strategy is that the robot moves repetitively through the same areas/paths. However, this is a quite reasonable assumption for service robots moving in relatively small offices or houses, that are the kind of environments in which we plan to use our system.

The proposed CML approach does not provides an exact position for the robot, but an approximation to it. However, this kind of rough information on the robot position is enough for most tasks, assuming that the low level behaviors of the robot controller are able to deal with local aspects of the environment (obstacles, doors, etc).

Due to the way in which we define the map, the map error will be small close to the areas where the map is started and growing for points far away from the origin. Actually, the error for a given map point p is is lower bounded by the error in odometry for a direct displacement from the origin of the map O to p. As a reference, our Nomad Scout robot can map an area of 20×20 meters with an accuracy below 1 meter. Since the error in odometry limits the area that we can map with a given accuracy, we would like to complement our localization system with additional odometric sensors (accelerometers, vision-based motion detection, etc) to determine more accurately the relative displacements of the robot. Another solution to enlarge the mapped area is to perform the CML in contiguous areas an then integrate the resulting sub-maps.

The results presented in this paper qualify the proposed approach as a very promising one, but much work has to be done to complete the system. A point that deserve further attention in our approach is the *map management*: how map points are efficiently stored and linked between them. A proper map structure would easy the task of keeping the map consistency, allowing an efficient back-propagation of the information obtained as the robot moves (introduction of new hypotheses, etc). More elaborated ways to perform the data association and the hypotheses pruning are also needed. Finally, another point we want to address in the near future is the integration of our CML system with a navigation module. For this purpose, we have to devise criterion to select the robot's movements so that accuracy in localization, accuracy in mapping, and efficiency in reaching the desired positions are simultaneously optimized.

ACKNOWLEDGMENTS

This work has been partially supported by the European (ITEA) project *"Ambience: Context Aware Environments for Ambient Services"*

REFERENCES

[1] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[2] I. Horswill, *Artificial Intelligence and Mobile Robots*. MIT Press, Cambridge, MA, 1998, ch. The Polly System, pp. 125–139.

[3] J. Leonard and H. Durrant-Whyte, "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[4] H. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.

[5] M. Jogan and A. Leonardis, "Robust Localization using Eigenspace of Spinning-Images," in *Proceedings of the IEEE Workshop on Omnidirectional Vision, Hilton Head Island, South Carolina*, 2000, pp. 37–44.

[6] B. Kröse, N. Vlassis, and R. Bunschoten, "Omnidirectional Vision for Appearance-based Robot Localization," *Lecture Notes in Computer Science*, vol. 2238, pp. 39–50, 2002.

[7] J. Crowley, F. Wallner, and B. Schiele, "Position Estimation Using Principal Components of Range Data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998, pp. 3121–3128.

[8] R. Sim and G. Dudek, "Comparing Image-based Localization Methods," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico*, 2003.

[9] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[10] H. Feder, J. Leonard, and C. Smith, "Adaptive Mobile Robot Navigation and Mapping," *International Journal of Robotics Research, Special Issue on Field and Service Robotics*, vol. 18, no. 7, pp. 650–668, 1999.

[11] S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Machine Learning*, vol. 31, pp. 29–53, 1998.

[12] P. E. Rybski, S. Roumeliotis, M. Gini, and N. Papanikolopoulos, "Appearance-Based Minimalistic Metric SLAM," in *Proceedings of the International Conference on Robotics and Intelligent Systems (IROS), Las Vegas, USA*, 2003, pp. 194–199.

[13] S. t. Hagen and B. Kröse, "Trajectory reconstruction for self-localization and map building," in *Proceedings of the IEEE International Conference on Robotics and Automation, Washington D.C., USA*, 2002, pp. 1796–1801.

[14] J. Gutmann and K. Konelige, "Incremental Mapping of Large Cyclic Environments," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*, 1999, pp. 318–325.

[15] H. Baltzakis and P. Trahanias, "Closing Multiple Loops while Mapping Features in Cyclic Environments," in *Proceedings of the International Conference on Robotics and Intelligent Systems (IROS), Las Vegas, USA*, 2003, pp. 717–722.

[16] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[17] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the Absolute Position of a Mobile Robot using Position Probability Grids," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1996, pp. 896–901.

[18] M. Pitt and N. Shephard, "Filtering Via Simulation: Auxiliary Particle Filters," *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, June 1999.

[19] N. Vlassis, B. Terwijn, and B. Kröse, "Auxiliary Particle Filter Robot Localization from High-Dimensional Sensor Observations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Washington D.C., USA*, May 2002, pp. 7–12.

[20] I. Cox and J. Leonard, "Modeling a Dynamic Environment Using a Multiple Hypothesis Approach," *Artificial Intelligence*, vol. 66, no. 2, pp. 311–344, 1994.

[21] K. O. Arras, J. A. Castellanos, and R. Siegwart, "Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2002, pp. 1371–1377.

[22] P. Jensfelt and S. Kristensen, "Active Global Localization for a Mobile Robot Using Multiple Hypothesis Tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 748–760, 2001.

[23] S. Kristensen and P. Jensfelt, "An Experimental Comparison of Localization Methods, the MHL Session," in *Proceedings of the International Conference on Robotics and Intelligent Systems (IROS), Las Vegas, USA*, 2003, pp. 992–997.

[24] J. Uhlmann, S. Julier, and M. Csorba, "Nondivergent Simultaneous Map Building and Localization Using Covariance Intersection," in *Proceedings of the SPIE Aerosense Conference, 3087*, 1997.

[25] H. Murase and S. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[26] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, 1990.

[27] R. Clarke, *Digital Compression of Still Images and Video*. Academic Press, 1995.