in: Proc. of the 2nd European Conference on Mobile Robots (ECMR 2005),
Ancona, Italy, pp. 206-211, stampalibri 2005 (ISBN 88-89177-187)

APPEARANCE-BASED CML APPROACH FOR A HOME STORE ENVIRONMENT

Alexander Koenig, Steffen Mueller, Horst-Michael Gross

Ilmenau Technical University
Department of Neuroinformatics and Cognitive Robotics
98684 Ilmenau, Germany
alexander.koenig@tu-ilmenau.de

ABSTRACT

Today the appearance-based robot localization in static environments is not really challenging. A number of methods, as for example Monte Carlo Localization, can solve this task more or less good. In this paper, we present a novel omnivision-based Concurrent Map-building and Localization (CML) approach which is able to localize a mobile robot in complex and dynamic environments and to adapt its environment model. From the modeling point of view, our method is closely related to the technique of Porta and Kroese [1] which also use a Gaussian-based multi-hypotheses tracker. We can show that our novel dynamic approach achieves localization results which are comparable to those of our previous particle-based localization approach. Furthermore, we demonstrate the problem of localization in changing environments with a static map and our solution for that. First results of home store experiments to build up a map while concurrently localize the robot are presented at the end of the paper.

1. INTRODUCTION AND MOTIVATION

Robust self-localization plays a central role in our long-term research project PERSES (PERsonal SErvice System) which aims to develop an interactive mobile shopping assistant which can autonomously guide its user within a home store [2]. To accommodate the challenges that arise from the specifics of this scenario and the characteristics of the operation area, a regularly structured, maze-like and populated environment, special emphasis is placed on vision-based methods for robot navigation. In our previous approach [3] (appearance-based Monte Carlo Localization), a static representation as map of the environment was developed, which was built up manually. The nodes of this map are labeled with visual observations extracted from the omnidirectional image and position information. Given this map, localization was realized by employing a Particle Filter to estimate the robot's state. The main drawback of this and other appearance-based approaches published for years is, however, that localization is only possible in manually mapped areas. Furthermore, the learned map is only valid as far as no important modifications of the operation area occur (bad assumption in a home store).

In recent years, more and more approaches were introduced, which are able to build up a map of environment and localize the robot concurrently. This also means, the robot continuously adapts its map if the operation area changes its appearance. As a consequence, a dynamic map for representation of a dynamic environment is necessary. Therefore, inspired by the work of Porta and Kroese [1],[4] and continuing our former work, an alternative technique was developed, which is able to perform an omnivision-based Concurrent Map-Building and Localization (CML) and to overcome this drawback. In this new approach, the static, pre-defined map of the operation area is replaced by an environment model which is obtained and continuously refined by the robot as it moves through the operation area. If the robot re-visits an already explored area, it can use the information previously stored to reduce the uncertainty of its position and to adapt its internal model. As extension to [1] and [4], our CML approach proposes improved fusion and learning methods and uses alternative observations which can be summarized as follows:

i) As observations and reference views, panoramic 360^o images are used, so-called omniviews, which describe a certain position in the environment under all possible heading directions of the robot. Therefore, this type of visual input is preferred to describe the appearance of a position.

ii) To determine the reference observations taken by the robot in positions visited earlier, often a crisp 1-of-N mapping from the current observation to the best fitting internal representation is used. Here, the new approach, however, employs a distributed coding, which uses a set of most similar reference observations to describe the current observation and to model the observation driven position hypothesis by a weighted superposition of the corresponding position estimations.

iii) While the model of Porta and Kroese can acquire new reference views and passively forget the positions of

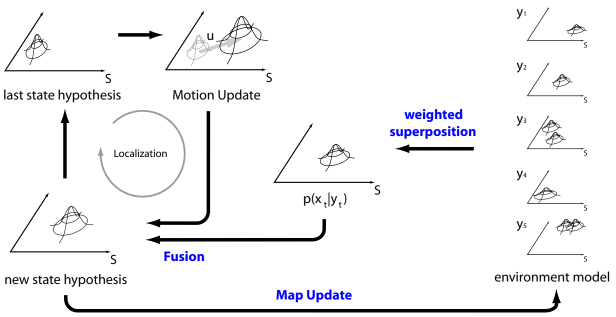

Fig. 1. General idea of our Concurrent Map-building and Localization (CML) approach from the modeling point of view closely related to [1],[4]. In both approaches, Mixtures of Gaussians (MoG) are used to represent both the belief of the robot's state $\mathbf{x}_t = (x_t, y_t, \phi_t)^T$ in the environment (left) and the environment model (right).

irrelevant ones to deal with dynamic effects, this new approach is in addition able to actively delete those views no more relevant for the environment model, e.g. due to permanent appearance changes at certain positions in the environment. This active forgetting is of central importance to keep the number of reference views under control.

The remainder of the paper is organized as follows: In section 2 the original CML and alternative or improved aspects of our omniview-based CML approach are explained. Section 2.4 describes the feature extraction from the omnicamera. Section 3 demonstrates CML's capability to localize the robot and to build up and adapt correct maps of the operation area. Localization results achieved with our static MCL [3] approach and this CML technique are discussed comparatively. At the end, the results are summarized and an outlook is given.

2. BASIC IDEA OF THE LOCALIZATION APPROACH

The basic CML approach of Porta and Kroese proposed recently [1],[4] is a promising technique to simultaneously build up an appearance-map of the environment and to use this map, still under construction, to improve the localization of the robot. In contrast to our previous MCL approach based on particle filters [3], the robot's state $\mathbf{x}_t$ is represented by a Mixture of Gaussians (MoG) (see Fig. 1, left). In general, a mixture $X_t = \{(\mu_t^i, \mathbf{C}_t^i, w_t^i) | i \in [1, N]\}$ is a set of partial hypotheses in form of single Gaussians with center μ_t^i and covariance matrix $\mathbf{C}_t^i$. Here, the weights w_t^i $(0 < w_t^i \leq 1)$ provide information on the certainty of the partial hypotheses. With that, the Belief of a robot's state is described as

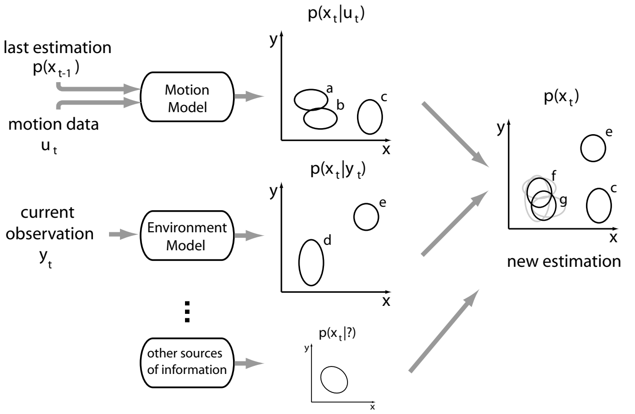

Fig. 2. Advanced fusion of several state hypotheses: state hypothesis generated by the last estimation and moved according to the motion data and the motion model (top, a-c), new observation-based state hypothesis from environment model (middle, d-e), and additional state hypotheses suggested by other position estimators (bottom) can be superimposed and merged to a new distribution coding the current state hypothesis (right). Mathematical details of this weighted merging are given in the text.

$$p(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{y}_t) = \sum_{i=1}^{N} w_t^i \phi(x | \mu_t^i, \mathbf{C}_t^i) \qquad (1)$$

with $\mathbf{x}_{t-1}$ as last state, $\mathbf{u}_t$ as motion information, and $\mathbf{y}_t$ as current observation. The current localization hypothesis $p(\mathbf{x}_t)$ is generated by a fusion of several state hypotheses coded as MoG. Given the last state hypothesis $p(\mathbf{x}_{t-1})$, the propagation of the motion data $\mathbf{u}_t$ by the motion model leads to a first hypothesis $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) = p(\mathbf{x}_t | \mathbf{u}_t)$ for the new state. In the following, this hypothesis has to be fused with another hypothesis, $p(\mathbf{x}_t | \mathbf{y}_t)$ which results from the environment model using the current observation $\mathbf{y}_t$ (or other state hypotheses from other source of information $p(\mathbf{x}_t | \ldots)$, see Fig. 1 and 2 middle). This requires an advanced fusion scheme in order to allow a weighted superposition of several information sources with different meaning, considering such aspect like reliability or stability of the hypotheses.

2.1. Generalized Scheme for Fusion of State Hypotheses

To fuse the two hypotheses generated by the motion model and the environment model (see Fig. 2, components a-e in the middle part), at first corresponding Gaussians in both hypotheses $a \in p(\mathbf{x}_t | \mathbf{u}_t)$ and $b \in p(\mathbf{x}_t | \mathbf{y}_t)$ representing similar positions are determined. As proposed by [4], this is done by a simple criterion based on the Mahalanobis distance in *State Space*:

$$D(a,b) = \left\{ \begin{array}{ll} 1 & : (\mu_a - \mu_b)\,(\mathbf{C}_a + \mathbf{C}_b)^{-1}\,(\mu_a - \mu_b) < \gamma \\ 0 & : else \end{array} \right. \tag{2}$$

If $D(a,b) = 1$, the two corresponding Gaussians are merged by Covariance Intersection. The objective of this fusion is to minimize the variance of the resulting distribution. Therefore, the resulting Gaussian gets the following parameters:

$$\mathbf{C}^i = \left((1-\alpha)\,\mathbf{C}_a^{-1} + \alpha\,\mathbf{C}_b^{-1}\right)^{-1} \tag{3}$$

$$\mu = \mathbf{C}^i\left[(1-\alpha)\,\mathbf{C}_a^{-1}\mu_a + \alpha\,\mathbf{C}_b^{-1}\mu_b\right] \tag{4}$$

The control parameter α determines which Gaussian dominates the merging result. Alternative to [4], we take the weights w_a and w_b of both Gaussians into account to realize a weighted fusion:

$$\alpha = \frac{w_b\; det(\mathbf{C}_a)}{w_b\; det(\mathbf{C}_a) + w_a\; det(\mathbf{C}_b)} \tag{5}$$

To determine the new weights w^i of the resulting Gaussians, a little bit effort is needed because one component can have more than one corresponding counterpart (e.g. component d in Fig. 2). For this purpose, for each Gaussian first the weights of all corresponding Gaussians within the allowed neighborhood (i.e., all components with $D(a,b) = 1$) are determined:

$$\hat{w}_a = \sum_{j|D(a,j)=1} w^j_{p(\mathbf{x}|\mathbf{y})} \tag{6}$$

$$\hat{w}_b = \sum_{j|D(j,b)=1} w^j_{p(\mathbf{x}|\mathbf{u})} \tag{7}$$

With that, the weight w^i of the Gaussianis given by

$$w^i = w_a\frac{w_b}{\hat{w}_a} + w_b\frac{w_a}{\hat{w}_b} \tag{8}$$

Gaussians in $p(\mathbf{x}_t|\mathbf{y}_t)$ and $p(\mathbf{x}_t|\mathbf{u}_t)$ without a counterpart in the other distribution (see Fig. 2, components c and e) are simply copied into the resulting MoG. This way, also completely new hypotheses can be integrated. During the final normalization of the resulting MoG, the weights of all unsupported components will be passively decreased. This way, their certainty is continuously reduced, and after a number of update cycles they can be deleted. To simplify the resulting MoG, overlapping adjacent Gaussians are merged to a single Gaussian.

Afterwards, the resulting MoG coding the new state hypothesis $p(\mathbf{x}_t)$ (see Fig. 1, bottom left or Fig. 2, right) is used for updating the environment model. Only unimodal hypotheses are employed for this update step, otherwise the reconstruction of states between two consecutive unimodal hypotheses takes place according to [4]. Experiments showed that this update must be delayed to avoid a positive feedback between state estimation and environment model and vice versa. For example, the changed $p(\mathbf{x}_t|\mathbf{y}_t)$ would directly influence $p(\mathbf{x}_t)$ after a short time, thereto, observation-state estimation pairs $(\mathbf{y}_t, p(\mathbf{x}_t))$ are stored in a short queue which is also used to buffer the time-consuming update operations of the environment model.

2.2. Improved Environment Model

The purpose of our improved environment model U is to allow an estimation of the current state $\mathbf{x}_t$ under the condition of the current observation $\mathbf{y}_t$ in a dynamic operation area with appearance variations. In our case, $\mathbf{y}_t$ represents the omnidirectional view in form of a describing feature vector (see 2.4). The whole *Observation Space O* is represented by a variable set of reference views $\mathbf{y}_i$. To each $\mathbf{y}_i$ a learned MoG X_i in the *State Space* is assigned.

$$U = \{(\mathbf{y}_i, X_i)\,,\; i = 1,\ldots,R\} \tag{9}$$

Each X_i is coding all those positions in the operation area the respective observation was captured before.

$$X_i = \{(w_{ij}, \eta_{ij}, \mu_{ij}, \mathbf{C}_{ij})\,|j = 1,\ldots,M_i\} \tag{10}$$

Every component j in X_i has a weight $w_{ij} > 0$ describing its relevance all over the model and for the fusion operation.

In extension to [4], we explicitly want to consider and model appearance variations and fluctuations occurring at identical positions within the environment. Therefore, we had to introduce another control parameter η_{ij} that counts the number of observations covered by a single Gaussian j in X_i.

Given the current observation $\mathbf{y}_t$, the localization hypothesis $p(\mathbf{x}_t|\mathbf{y}_t)$ can be simply generated by finding the most similar reference-view $\mathbf{y}_i$ in the model U, as proposed by [4]. However, to achieve a smoother approximation of the hypothesis $p(\mathbf{x}_t|\mathbf{y}_t)$, in our approach we realize a weighted superposition of several position hypotheses. To be precise, we superimpose the position hypotheses X_i of all that reference views $\mathbf{y}_i$ which are similar to the current observation $\mathbf{y}_t$ to minimum degree. As illustrated in Fig. 3-top left, only those reference-views are considered in this superposition whose distance to the current observation is lower than the limit distanceE^{max}. Therefore, a similarity measure $S(\mathbf{y}_t, \mathbf{y}_i)$ was introduced which is 1.0 for a perfect matching and continuously decreases to zero up to the E^{max}. The final state hypothesis $p(\mathbf{x}_t|\mathbf{y}_t)$ is computed as weighted sum over all X_i whereas the individual weights w_{ij} of the Gaussians are multiplied by the similarity values $S(\mathbf{y}_t, \mathbf{y}_i)$ of the respective reference views.

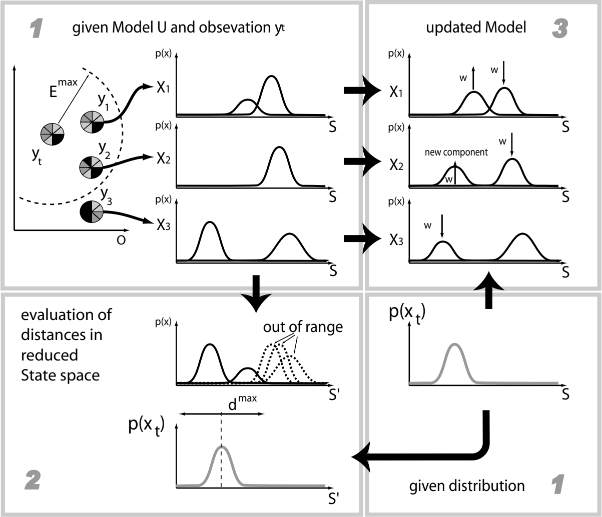

Fig. 3. Schematic illustration of the weight updating in the Environment Model in three phases: (**Phase 1**) given are the old model and the new observation $\mathbf{y}_t$ to be inserted (top left) and its current position estimation $p(\mathbf{x}_t)$ (bottom right); (**Phase 2**) at first, similarities between the Gaussians in X_i and $p(\mathbf{x}_t)$ are determined; (**Phase 3**) the similarity of observation $\mathbf{y}_t$ and known reference views $\mathbf{y}_i$ decides in what way the certainty weights w_{ij} of the Gaussians have to be adapted.

2.3. Updating the Environment Model

While using the model for generation of state hypotheses, new observations are made that must be integrated. In the simplest case, the model is only updated by pairs of observation $\mathbf{y}_t$ and unimodal state distribution $p(\mathbf{x}_t)$, i.e. a single Gaussian $\phi(\mathbf{x}|\mu_t, \mathbf{C}_t)$. A dynamic environment and the arising stability requirements, however, need a more complex update regime that has to take into consideration the following premises: i) one observation should only be generated by one position in the area and ii) each position does only show one appearance at a time. Therefore, the update of the environment model is typically carried out in three phases.

1) Insertion of a new reference view: If the feature distance between the current observation and all learned reference views is larger than the limit distance E^{min}, the current observation $\mathbf{y}_t$ has to be stored as a new reference view $\mathbf{y}_n$.

2) Update of the Mixtures of Gaussians: A new Gaussian representing the current state hypothesis $p(\mathbf{x}_t)$ is added to the already existing Gaussian Mixtures X_i using the similarity values $S(\mathbf{y}_t, \mathbf{y}_i)$ mentioned above as gain control. The basic idea behind this update step is as follows: since each X_i is the result of all former observations, a balanced insertion of new Gaussians has to be realized. Therefore, if a new component k has to be inserted into X_i, the η_{ij} of the already existing components need to be taken into account. The resulting new Gaussian k approximates the sum of the two density functions X_i and $p(x_t)$ weighted by η_{ij}. Because η_{ik} is increased for each fitting component, the influence of a single update step decreases with time, and the MoG stabilizes gradually.

3) Adaptation of the Gaussian's weights: Now, the certainty weights w_{ij} of the Gaussians are updated according to the premises introduced above. For this, first the feature similarities $S(\mathbf{y}_t, \mathbf{y}_i)$ in the *Observation Space O* are determined (see Fig. 3 - top left). Furthermore, the similarity of each Gaussian j in X_i to the single Gaussian describing the unimodal current state $p(\mathbf{x}_t)$ is determined by means of the Mahalanobis distance (see Fig. 3 bottom left).

Now, the weights w_{ij} of every Gaussian j in each X_i are adapted, if one of the following conditions is fulfilled:

1. *Weight increasing:* If $\mathbf{y}_t$ equals $\mathbf{y}_i$ and the current position estimation $p(\mathbf{x}_t)$ resembles the Gaussian j then the weight w_{ij} of this Gaussian is increased (see Fig. 3, top right - left Gaussian in X_1).

2. *Passive forgetting:* If $\mathbf{y}_t$ is matching reference observation $\mathbf{y}_i$ but the considered Gaussian j of X_i does not match to $p(\mathbf{x}_t)$, then the weight of the respective Gaussian has to be decreased (see Fig. 3, top right - right Gaussians in X_1 and X_2).

3. *Active forgetting:* If reference observation $\mathbf{y}_i$ doesn't match $\mathbf{y}_t$ but has a component at the same (x, y)-position as $p(\mathbf{x}_t)$ then the respective weight w_{ij} has to be decreased too, to gradually forget former observations at this position, which will not appear again (see Fig. 3, top right - left Gaussian in X_3).

After updating, Gaussians with too low weights are removed in X_i and, as consequence of this, all reference views with empty X_i are deleted. This way, the last update rule realizes a limited number of Gaussians in a restricted local area. Moreover, it guarantees, that old, irrelevant observations at those positions with variable appearances can be replaced by new ones. By means of this forgetting, the complexity of the algorithm, which is determined by the number of reference views, is linear in the area the robot is operating in. If a former observation, however, should appear again at the same position, it can be inserted as a new observation into the environment model again.

2.4. Extraction of Omni-Features

To allow a comparison of images, besides the image representation $\mathbf{y}$, a similarity-function $S(\mathbf{y}_1, \mathbf{y}_2)$ must be given. This quantifies the differences between the two images $\mathbf{y}_1$ and $\mathbf{y}_2$. So, similar omniviews should result in a S-value near one, while differing views deliver low similarities. For robust and exact localization, there are different aspects, which characterize a good feature representation and similarity-function. Another important aspect is, that the CML approach for the activation of the position hypotheses uses a rotation invariant comparison of images (see below), but for the estimation of the robot's orientation the rotation angle between the current observation and the respective reference observation is required.

For the last point, according to our former MCL approach [3] a RGB-based representation of the omniview was used to get comparable experimental results between CML and MCL localization. Therefore, by means of subdividing the omnidirectional image into radial segments (usually 36 sections), the mean RGB-value of pixels inside each segment is used as a set of features (see [2]). Given these features, two views can be directly compared by using the sum of absolute differences of intensities in each segment and each channel as an similarity-value. To determine a rotation invariant similarity all possible orientations have to be considered, and the maximum similarity is used. This operation is needed to find similar observations in the map. The other important part is the compution of the rotation angle between two images which is determined by a cross correlation. To compute this more efficiently, the operation is done in the frequency domain. This operation is required to correct the orientation component of participating distributions while extracting the hypothesis $p(\mathbf{x}|\mathbf{y})$.

In the next section, our CML approach will be investigated in several localization experiments. At first, the general capability to globally localize the robot has to be shown. After that, the CML localization approach is examined in view of adapting/non-adapting the map under varying appearance of the environment (dynamic changes). Finally, a map will be built up without any kind of position reference information i.e. the map building starts with an empty map.

3. OPERATION AREA AND EXPERIMENTS

In our test environment, a typical home store, many dynamic effects occur. All data for the analysis were recorded in the home store under realistic conditions, i.e. customer and employees walked through the operation area, good shelves were rearranged, and other dynamic changes (illumination, ...) happened.The data contain the images and motion commands the robot has obtained while it moved through the home store. The main part of the data was used to build up both a CML and MCL map and the remainder to carry out the localization experiments. To increase the dynamic effects, two data sets with a time difference of four weeks were recorded, which were used for the second experiment. Each data set contains more than 100,000 images and odometric data. For this, the robot was moved about 2,000 meters through the operation area by joysticking. The maps have a size of 50 x 80 meters. The driven test routes typically have a length of about 300 meters.

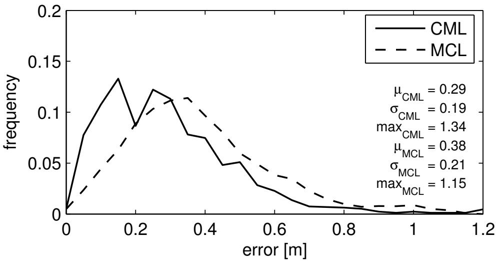

Fig. 4. Experimental results for the CML and MCL localization approaches in a static environment illustrated as histogram of errors. The histogram gives a more precise impression of the error distribution than absolute values μ and σ. A stronger concentration on the left side means more small errors, therefore a smaller average error μ. In contrast, plots stretching out to the right contain more high errors, which corresponds to a higher average position error. Both localization algorithms achieve a comparable accuracy in a static environments.

The first experiment was carried out to determine the capability of CML approach to track the robot in a static environment. At the beginning, the robot's state was initialized at a known position, but with uncertainty. More than 3,000 state estimations were made by the localization approach along the test routes. Based on this, the mean localization error $\mu(e)$, the variance of the error $\sigma(e)$ and the maximum error were computed. In Figure 4 the results for the CML and our previous localization method [3], the omnivision-based Monte Carlo Localization, are shown. The mean localization error for the MCL approach is 38cm. The accuracy for CML is at an average of 29cm. It can be seen that both localization methods produce a satisfying position estimation on a static ”perfect” map, this means, the maps were built up off-line with ”perfect” corrected odometric data. Despite the different coding principle, the new MoG-based approach presented here, shows a similar accuracy as the particle-based algorithm (MCL). But, the computation time for a estimation step is clearly higher than a step of a particle-based approach. With some practical improvements (using only local parts of environment map), an estimation step needs about 1 second on a 2GHz CPU.

The next experiment was executed to show, that dynamic environments require a dynamic map, i.e. a map which can be adapted. At first, the ”old” predefined map based on the first experiment was used to localize the robot on data recorded four weeks later. During this time, many locations in the home store were rearranged which caused an higher localization error (see Fig. 5). Here, the average error was 111cm. After that, the capability of the CML algorithm to adapt the learned map was activated. Now, the mean localization error could be decreased to 58cm by using the adapted map. Due to the changes in the environment, in the experiment using the none-adapted map numerous localization failures produce high maximum localization errors (see Fig. 5) that influence the average error negatively. The failures happened if the current observation was similar to more than one reference observation with different positions in the map. Therefore, the localization hypothesis sometimes jumped to another location. The experiment clearly illustrates that an adapting map is necessary in dynamic environments.

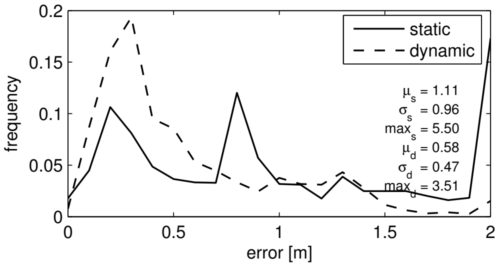

Fig. 5. Experimental results for the CML localization in a changing environment. The diagram shows that the localization error based on a static map in a dynamic environment is significantly higher (→ static), whereby a dynamic map is able to improve the localization (→ dynamic). A stronger concentration on the left side means more small errors, and therefore a smaller average error μ.

In the last experiment, it was tried to concurrently localize the robot and to build up the map. In contrast to the first experiment, the map was built online and without ”perfect” odometric data and the algorithm started with an empty map. Therefore, it is important that the CML algorithm can recognize already visited areas to create a stable and consistent map. Figure 6 (left) shows the navigation path determined by the CML approach for a local area. The robot built up the map while it was driven on three days through the environment. In this area of about 30m by 30m, the average localization error and also the environment model was improved with every day (see Fig. 6, right). The end, an average localization error of about 40 cm was achieved. This experiment shows that it is possible to localize and to build up a map concurrently with the omnivision-based CML approach. But it is relatively time-consuming to get a consistent map because of the necessity of a repeated observation of the same operation area. For this map, the robot covered a distance of about 7,000 meters.

4. CONCLUSION AND FUTURE WORK

It could be shown that the MoG-based CML algorithm behaves as good as the well known particle-based MCL approach. Furthermore, the necessity for an adapting map to

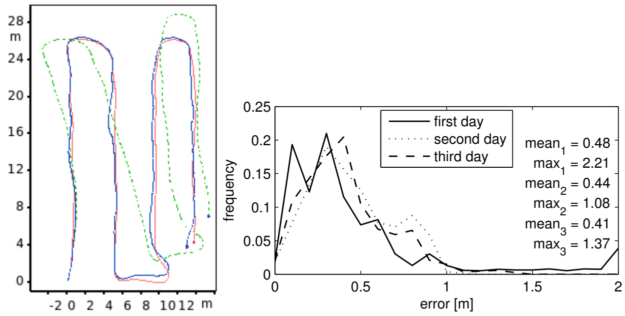

Fig. 6. Experimental results of our long term experiment in the home store. (Left) As reference for visualization good shelves (grey rectangles), true path (red/light grey), estimated path (blue/dark grey) and odometric data (green/dotted) are shown. (Right) The error histogram clarified the improvements made by the CML algorithm with every new data (from day to day). The average error is improved from 48 cm to 41 cm (maximal error 221 cm to 137 cm)

represent changing environments could be demonstrated by the second experiment, where the localization error could be decreased. The third experiment showed first encouraging results to online build up a map of a large operation area. In future experiments, we plan to build larger maps (more hallways and shelves) over a longer period to show that a stable and consistent map with converging localization errors is possible, what could be at test partially demonstrated by the last experiment.

5. REFERENCES

[1] J. M. Porta and B. J. A. Kroese, “Appearance-based concurrent map building and localization,” in *Int. Conf. on Intelligent Autonomous Systems*, 2004, p. 10221029.

[2] H.-M. Gross, A. Koenig, H.-J. Boehme, and C. Schroeter, “Vision-based monte carlo self-localization for a mobile service robot acting as shopping assistant in a home store,” in *Proceedings of IROS’2002, IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland*, October 2002, pp. 256–262.

[3] H.-M. Gross, C. Koenig, A. Schroeter, and H.-J. Boehme, “Omnivision-based probabilistic self-localization for a mobile shopping assistant continued,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 1505–1511.

[4] J. M. Porta and B. J.A. Kroese, “Appearance-based Concurrent Map Building and Localization using a Multi-Hypotheses Tracker,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004, pp. 3424–3429.