Robust offline topological map estimation using visual loop closures

Dimitrios Kosmopoulos
TEI of Crete
71500 Heraklion
Greece
dkosmo@ieee.org

Ilias Maglogiannis
University of Piraeus
18532 Piraeus
Greece
imaglo@unipi.gr

Fillia Makedon
University of Texas at Arlington
76019 TX, Arlington
makedon@uta.edu

ABSTRACT

A framework employing the Student-t pdf is introduced for offline map estimation and robot localization using visual loop closures. The framework uses the Student-t pdf (a) as an observation model of a Hidden Markov Model to represent a topological map (b) to represent the robot motion model. The map and the motion model are calculated in an expectation maximization (EM) framework. We show that the estimator converges at linear time and that the provided accuracy is higher compared to using a conventional Gaussian mixture pdf, due to higher noise resiliency, as well as compared to using a fixed robot motion model. The task is assisted by unsupervised landmark definition through the EM-based clustering of the observations and by scene representation using the complex Zernike moments, which provide rich rotation-invariant information. The validity of the method has been verified experimentally using the input from an omnidirectional camera.

Categories and Subject Descriptors

Artificial Intelligence [**Computer Vision**]: Computer Vision Tasks—*Vision for robotics*; Machine Learning [**Machine learning approaches**]: Learning in probabilistic graphical models— *Maximum likelihood modeling*

Keywords

Hidden Markov Model, Expectation Maximization, Zernike Moments

1. INTRODUCTION

Map acquisition for mobile robots is a research field rapidly gaining momentum over the last few years. This is mainly due to the fundamental applications of maps in autonomous robot navigation and specifically in mission and path planning as well as in localization. Such robotic tasks are critical in applications such as assistive robots for sensitive social groups (e.g., elderly, patients, etc), see for example [15], [5].

PETRAE '13 Rhodes, Greece

As pointed out by researchers the problem of mapping and localization is a sort of chicken and egg problem. To determine the location of observed objects the robot must know where it is and to determine where it is it must know the location of the observed objects. Therefore the two problems are typically treated simultaneously (Simultaneous Localization And Mapping - SLAM).

In this work we investigate the applicability of the Student-t pdf towards solving a SLAM problem using an offline method. The Student-t pdf has an advantage compared to the conventional Gaussian pdf, due to its higher noise resiliency. Here we employ it to calculate a map after the tour is finished, so that the same measurements can be revisited to obtain incrementally the best possible map.

We exploit the loop closures to calculate the map and the robot position simultaneously with the robot motion model, which may lead to high position accuracy. The implementation associates the visual input and the odometric data in an expectation maximization framework. The task is assisted by unsupervised landmark definition and by scene representation using the complex Zernike moments, which provide rich rotation-invariant information.

The rest of this work is organized as follows. In the next section we present the related work to highlight the utility of the proposed method. In section 5 we present the core of our learning framework. Section 4 introduces the proposed method for scene representation. In section 6 we present the experimental results and section 7 concludes this work.

2. RELATED WORK

The problem of SLAM has attracted many researchers in the past. It was initially theoretically approached by [10] and [21], which gave the foundations for the statistical processing of the landmarks. It was shown later that the estimation about the landmarks are correlated due to the common systematic error in the position estimation of the robot [22]. This correlation is actually exploitable to solve the problem.

Very popular are the filter-based methods. Some of the initial approaches used the extended Kalman filter to track maps. Examples of such methods are given in [9] and [13]. Other more advanced methods employ particle filters, which make less assumptions about the linearity and the noise model. Such methods are the FastSLAM [17], as well as the DP-SLAM [11]. These methods are executed online. At each moment in time, online methods can use all data, up to that moment in time. In other words, they are not "allowed" to see into the future. An obvious problem of those methods is the map dimensionality, which may undermine real time performance.

Another class of methods is the offline map estimation. In contrast to online methods, the offline ones have the possibility to consider all measurements at any time, which allows re-estimation of a map until a good solution is found. Characteristic approaches are given in [25], [24]. Due to the offline nature of these methods the real time execution is not a strict requirement and thus the high dimensionality is not prohibitive. The application presented here belongs to this class.

The topological maps represent the environment as a graph; the nodes represent specific landmarks which may have a semantic meaning, e.g., door, pathway etc. The edges of the graph describe the transition between the landmarks. The utility of the topological maps stems from their ability to represent semantic concepts and are therefore often used in a complementary fashion to a metric map see, e.g., [12], [26]. Topological maps can be learned after extracting a metric map (e.g., [23]) or directly (e.g., [20]), which is more efficient and does not require the existence of a metric map. We follow the latter approach.

The topological map graph can be well represented by a hidden Markov model (HMM). This has been exploited in [20] and [12]. The states of the HMM represent the nodes of the graph and the edges are represented by the transition probabilities. In these works the observations (sensor readings) are represented by Gaussian mixture models (GMMs). However, such models are rather sensitive to noise. In our recent work we have demonstrated that it is possible to substitute the GMMs with Student-t mixture models (SMMs) and thus endow the HMM with high tolerance to noise [7], [6]. This is very important for SLAM applications, where high amount of noise is expected due to imperfections in the data acquisition procedure, as well as due to dynamic environments.

In our experiments we rely on the detection of loop closures, which has been used in the past for accurate estimation of topological maps (see e.g., [2], [14]). By loop closure detection we mean the problem of correctly asserting that a robot has returned to a previously visited area. Here we detect loop closures and we exploit them to estimate both the topological map and the robot motion model simultaneously.

The idea of incorporating the robot motion model in order to extract a map more accurately has been highlighted in [20], where an enhanced HMM includes the odometric relations between the states. This work differs from [20] in that we define the states differently by using orientation-invariant features extracted from omnidirectional images. This way each location is associated with only one state, which is closer to human perception. Moreover, we do not explicitly model the transitions between the states using an odometry model, on the contrary we have a single model for the whole robot path. This is a result of the different state definition, but also is simpler to learn.

The problem of place recognition is closely related to this approach. Typical visual cues that are employed in literature are either holistic like the color histogram [3] or based on local features like the SIFT see, e.g., [28], [4]. Here we use holistic features, namely the Zernike moments, however the proposed framework is not limited by the loop closure detection method. Therefore other approaches such as the bag-of-words (see e.g., [8]) can also be applied.

A side product of using HMMs to represent a map is the clustering into states/landmarks, through training. The idea of unsupervised learning to represent places of similar appearance has been successfully employed in previous works see, e.g., [28].

Considering the above mentioned work, we implemented for the purposes of our experiments a testbed similar to [24] but with several additional features, which include, (a) the definition of landmarks in an unsupervised way, (b) the use of an adaptive robot motion model and (c) the exploitation of loop closures to correct the odometry error. However, the main motivation behind this research was not just to improve that method, but rather to use its paradigm to demonstrate the utility of Student-t based models in SLAM applications and their superiority compared to the Gaussian approaches. This can have impact on SLAM applications that employ HMMs to build a map or that employ a probabilistic model to represent the odometry errors.

3. DEFINITIONS

In this section we describe and justify the tools that we use for the map probability calculation, for the dynamic model and the nodes.

3.1 Map probability

In the following similarly to [24] we are going to present the learning framework for offline mapping, i.e., after the robot has already finished with the mapping tour. The data available for mapping is of the form

$$d = \{o_1, u_1, ..., u_{T-1}, o_T\} \tag{1}$$

where o_t and u_t denote the observation and the robot control command in time t respectively. Here the problem of mapping boils down to finding the most likely map $\hat{m}$ given the data according to:

$$\hat{m} = \arg\max p(m|d) \tag{2}$$

or equivalently according to (3) [24] (see next page). As stated in the same work the calculation is computationally challenging because finding the most likely map involves search in the space of all maps and integrating over all possible locations at all points in time.

3.2 Robot motion model

Here we propose to additionally include in (3) the robot motion model, or equivalently a model for the odometry error, which may provide some additional accuracy in the calculations. Due to wheel slippage and measurement error in rotation and distance between wheels, there is systematic and non systematic error in the estimation of position using odometry.

Assuming small intervals between two position estimations the robot motion model, which associates the current pose with the previous one and the control command can be represented by a mixture of probability density functions, as follows:

$$p(x_t|u_{t-1}, x_{t-1}) = \sum_{i=1}^{k} c_i p(\mu_{ei} - x_t + x_{t-1} + u_{t-1}, \Sigma_{ei}) \tag{4}$$

$$\hat{m} = \arg\max_{m} \int ... \int \prod_{t=1}^{T} p(o_t|m, x_t) \prod_{t=1}^{T-1} p(x_{t+1}|u_t, x_t) dx_1 ... dx_T \tag{3}$$

$$c_i > 0, \quad \sum_{i=1}^{k} c_i = 1 \tag{5}$$

where x_t is the position in time t, k the number of components; μ_{ei}, Σ_{ei} are the mean and covariance of the error for the i-th mixture component and c_i the component priors. By using the appropriate number of components, depending on the specific environment, it is possible to model both systematic and non systematic errors.

Here we propose to use as pdf in the mixture in (4) the Student-t. The Student-t distribution with mean vector μ, positive definite inner product matrix Σ, and ν degrees of freedom is given by:

$$p(\mathbf{y}_t; \mu, \Sigma, \nu) = \frac{\Gamma\left(\frac{\nu+p}{2}\right) |\Sigma|^{-1/2} (\pi\nu)^{-p/2}}{\Gamma(\nu/2)\{1 + d(\mathbf{y}_t, \mu; \Sigma)/\nu\}^{(\nu+p)/2}} \tag{6}$$

where p is the dimensionality of $\mathbf{y}_t$, d is the squared Mahalanobis distance between $\mathbf{y}_t$, and μ with covariance matrix Σ, and $\Gamma(s)$ is the Gamma function. The Gaussian pdf is the specific case of a Student-t with ν=∞. As demonstrated in fig. 1 the Student-t pdf has heavier tails and therefore is able to incorporate outliers instead of being corrupted by them.

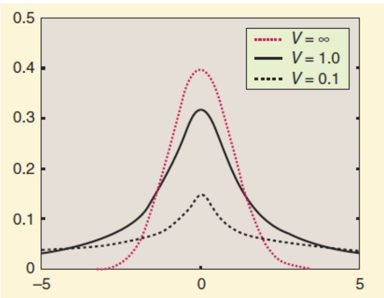


Figure 1: The Student-t distribution for various ν values. For $\nu \rightarrow \infty$ we yield the Gaussian distribution.

3.3 Topology description through an HMM

For the representation of a topological map we use here the HMM, which is a popular framework for modeling time series. Here we use its capability to separate the observation from the states.

An HMM consists of states, transitions, observations and probabilistic behavior, and is formally defined as a tuple $\lambda = \langle Q, O, A, B, \pi, \rangle$ satisfying the following conditions:

- $Q = \{q_1, ..., q_N\}$ is a finite set of N states.
- $O = \{o_1, ...; o_T\}$ is a finite set of T possible observation values.
- A is the transition matrix, which represents the transition probabilities between states.
- B is the observation matrix, which represents the observation probability given the state.
- π represents the probability of each state at the beginning of the sequence.

More specifically the states q_t are considered to be the labels that describe some typical views that can be observed through the sensors in time t, e.g., pathways, junctions, doors etc. In big buildings it is very common that some views are repeated and this justifies having a set of landmarks that cover all possible semantic descriptions. All the nodes of a topological map, can be described by such labels-states. The transition probabilities between the states of the HMM are actually the probabilities to go from one label to another.

The observations o_t extracted from the sensor readings at time t represent the visible landmarks, and will be further explained in section 4. Here we propose a Student-t mixture model to represent the observations at each HMM state.

The probability of being in state i in time step t is given by the following:

$$p(q_t = i|O, \lambda) = \frac{a_{ht}(i)\beta_{ht}(i)}{\sum_{i=1}^{N} a_{ht}(i)\beta_{ht}(i)} \equiv \gamma_t(i) \tag{7}$$

where a_{ht}, β_{ht} are the forward and backward variables of the HMM, which are calculated in a dynamic programming fashion during training [18] and are given by:

$$a_{ht}(i) = p(o_1, ...o_t, q_t = i|\lambda) \tag{8}$$

$$\beta_{ht}(i) = p(o_{t+1}, ...o_T, q_t = i|\lambda) \tag{9}$$

Given that we have associated each state with a landmark, actually the equation (7) calculated the probability of seeing the landmark i in time t given the HMM and the observation sequence.

4. LANDMARK REPRESENTATION AND LOOP CLOSURE DETECTION

By detecting loop closure we aim to assert that a robot has returned to a previously visited area. This assertion has to be independent of robot orientation, i.e., only the position should be considered. To this end omnidirectional vision is an attractive option. The visual observations (landmarks) should be therefore described by vectors which are not affected by orientation variations. These observations can be used in the following for the definition of a topological map.

The complex Zernike moments lend themselves as an attractive representation option, especially for circular image regions such as those provided by omnidirectional vision [16]. They are invariant to rotation and have some interesting properties such us noise resilience, lack of information redundancy due to their orthogonality and high reconstruction capability.

We should not here that the proposed framework is not limited by the use of the Zernike moments to represent visual landmarks. Other methods such as bag-of-words [8] or Fourier signatures are applicable as well.

The complex Zernike moments of order p are defined on an image f as:

$$A_{pq} = \frac{p+1}{\pi} \int_0^1 \int_{-\pi}^{\pi} R_{pq}(r) e^{-jq\theta} f(r,\theta) r dr d\theta \qquad (10)$$

where $r = \sqrt{x^2+y^2}$, and $\theta = \tan^{-1}(\mathrm{y/x})$ and $-1 < x, y < 1$ (x,y are the transformed image coordinates, with respect to the image center) and:

$$R_{pq}(r) = \sum_{s=0}^{\frac{p-q}{2}} (-1)^s \frac{(p-s)!}{s!(\frac{p+q}{2}-s)!(\frac{p-q}{2}-s)!} r^{p-2s} \qquad (11)$$

where $p - q = even$ and $0 \le q \le p$. Moments of low order hold the coarse information while the ones of higher order hold the fine details. However, the more detailed the region representation is, the more processing power will be required, and thus a trade-off has to be reached considering the specific application requirements.

The loop closures detection is crucial for the unsupervised estimation of a topological map. The loop closures are detected simply as the locations where the distance between two vectors is very low according to the representation that we have selected in the previous section. For this purpose we have used the Euclidean distance and as possible loop closures we select the locations with very small mutual distance. Since there are many such locations in a local neighborhood we select the vectors that present a local minimum. We have seen experimentally that the behavior of the error is locally linear provided that the frame rate is relatively high and the robot does not move too fast. The threshold is experimentally defined by observing several actual loop closure cases.

An issue about the loop closure detection is what happens when the environment is quite similar in two very different locations (i.e., two very similarly looking offices). This will generally result in the false detection of a loop closure. Integrating some upper bound to the average odometry error can most of the times resovle that in case that the similar looking places are in very different locations.

5. LEARNING FRAMEWORK

As soon as a loop closure is detected, we can assume that the initial and the final position practically coincide. In the following we demonstrate how to use this information to learn efficiently a map using an expectation-maximization EM framework, based on the definitions in the previous sections. The training of a Student-t HMM can be found in our previous work [7].

5.1 E-step

The probability of being in x_t given the observation/control sequence and given the current map can be efficiently calculated by [24]:

$$p(x_t|d,m) = n \cdot \underbrace{p(x_t|o_1,...,u_{t-1},o_t,m)}_{a(x_t)} \cdot \underbrace{p(x_t|u_t,...,u_{T-1},o_T)}_{\beta(x_t)} \qquad (12)$$

where $a(x_t)$,$\beta(x_t)$, are computed separately; the former is computed forward in time and the latter is computed backwards in time as given in the following formulas:

$$a(x_1) = p(x_1|o_1,m) = \begin{cases} 1 & if \quad x_1 = (u_0,v_0) \\ 0 & if \quad x_1 \neq (u_0,v_0) \end{cases} \qquad (13)$$

$$a(x_t) = n \cdot p(o_t|x_t,m) \int p(x_t|u_{t-1},x_{t-1}) \cdot a(x_{t-1}) dx_{t-1} \qquad (14)$$

$$\beta(x_t) = \int p(x_{t+1}|u_t,x_t) p(o_{t+1}|x_{t+1},m) \beta(x_{t+1}) dx_{t+1} \qquad (15)$$

$$\beta(x_T) = \begin{cases} 1 & if \quad x_T = (u_0,v_0) \\ 0 & if \quad x_T \neq (u_0,v_0) \end{cases} \qquad (16)$$

where (v_0,u_0) is the initial robot position, at the beginning of the loop, which coincides with the robot position after closing the loop.

The expected position of the robot in time step t is calculated by:

$$E(x_t) = n \int x_t p(x_t|d,m) dx_t \qquad (17)$$

where n is a normalizing constant.

We observe that in time t the expected odometry error e_t is given by:

$$e_t = E(x_t) - E(x_{t-1}) - u_{t-1} \qquad (18)$$

5.2 M-step

In the maximization step we compute the most likely map based on the probabilities computed in the E-step. Given the factors $\alpha(x_t)$, $\beta(x_t)$ from eq. (13) - (16), the probability $\gamma_t(i)$ of being in state i in time t given by eq. (7) and the probability $p(o_t|q_t = i)$ of the observation given the landmark (obtained from the automatically calculated state model of the HMM described in sub-section 3.3) we propose to calculate the following probabilities.

The probability that in position X the landmark is of type i given the HMM λ will be:

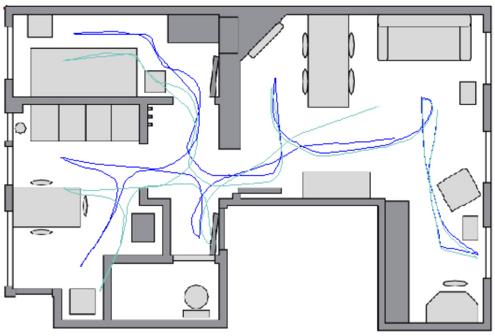
$$p(q_t = i|X,\lambda) = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot a(x_t = X) \cdot \beta(x_t = X)}{\sum_{t=1}^{T} a(x_t = X) \cdot \beta(x_t = X)} \qquad (19)$$

The probability that we will observe o_t when we are in position X:

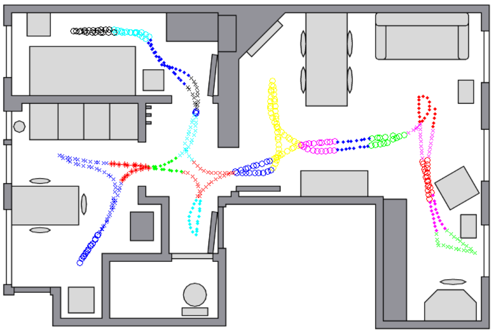
$$p(o_t|X,m) = \sum_i p(o_t|q_t = i) p(q_t = i|X,\lambda) \qquad (20)$$

Furthermore, we propose to model the pdf of the e_t is modeled using a standard Student-t mixture model using the results from the eq. (18). The enhanced error model is actually the probability $p(x_{t+1}|u_t,x_t)$, which is subsequently used in equations (14), (15).

The Student-t mixture parameters can be estimated using a standard expectation maximization approach (see e.g., [1]).



(a)



(b)

Figure 2: 2D map of the home environment where the data was recorded. (a) With dark color the ground truth path and with light color the perceived through odometry (b) The extracted clusters of landmarks denoted by markers of different colors and shapes

6. EXPERIMENTS

We have evaluated the proposed method experimentally using a publicly available dataset. More specifically we used the dataset produced by the project COGNIRON [27]. The acquisition of the data set took place in a home environment, see fig.2. The mobile robot was driven around by tele-operation to collect the data. The following sensors were used:

- Omnidirectional camera using a camera with a hyperbolic mirror. The image resolution was 1024×768.
- A SICK-laser scanner (LMS-200) was used to record range scans at the front of the robot. The scanner was running in millimeter precision, 0.5 degree angular resolution over 180 degrees and had approximately 8 meter maximum range.
- Odometry. On average 12 odometry measurements per second were taken.

The main goal in our experiments was to prove that the proposed model has higher tolerance to noise than the conventional Gaussian approach. We also verified:

- the convergence of the iterative method
- the accuracy of the produced map
- the ability to represent the observations

The observation vectors were composed of the norms of the complex Zernike moments, which were selected due to their rotational invariance. The order of the moments used was up to 17, yielding vectors of size 90. We downscaled the original images to 307×230 for higher efficiency. The circular integration was calculated starting from a radius of 80 pixels up to a radius of 105 pixels. That way we removed the circular disk in the image center which provided a constant view due to camera configuration and thus no useful information. That choice provided an acceptable reconstruction of the omnidirectional images. See for example fig.3 for some sample reconstructions using the proposed representation. Noise from blurred images as well as from people moving around was naturally expected to produce outliers and to corrupt the observations.

We have selected N=21 states (nodes) to train the HMM using the BIC criterion [19]. The observations that looked similar were naturally clustered around neighboring positions. This becomes obvious by observing the fig.2b. Each cluster of observations can be replaced by a node in a topological map (e.g., the mean or medoid of the observation vectors belonging to the cluster).

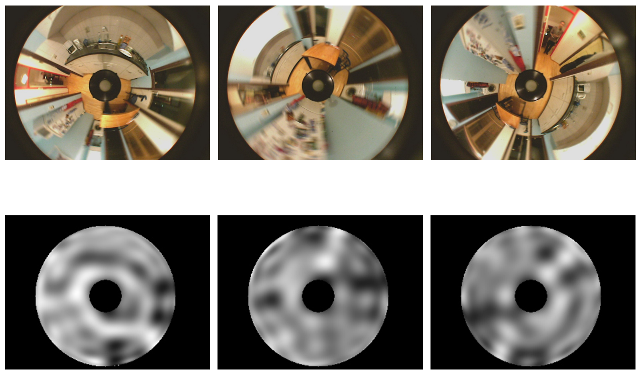


Figure 3: Representation using the Zernike moments. Sample image reconstructions using moments of 17th order provide rough yet discriminative approximations

We were able to detect automatically about 50% of the loop closures without false positives simply by using the feature vector distance as described in section 4. Other techniques (e.g., [8], [2]) can be combined with this framework to detect more loop closures, avoiding at the same time false positives that could corrupt the produced map.

We examined the framework with regard to its convergence properties and its accuracy compared to ground truth. For this purpose we utilized the measurements obtained by the laser range scanner, which is the most accurate measure available in the COGNIRON dataset. The accuracy was

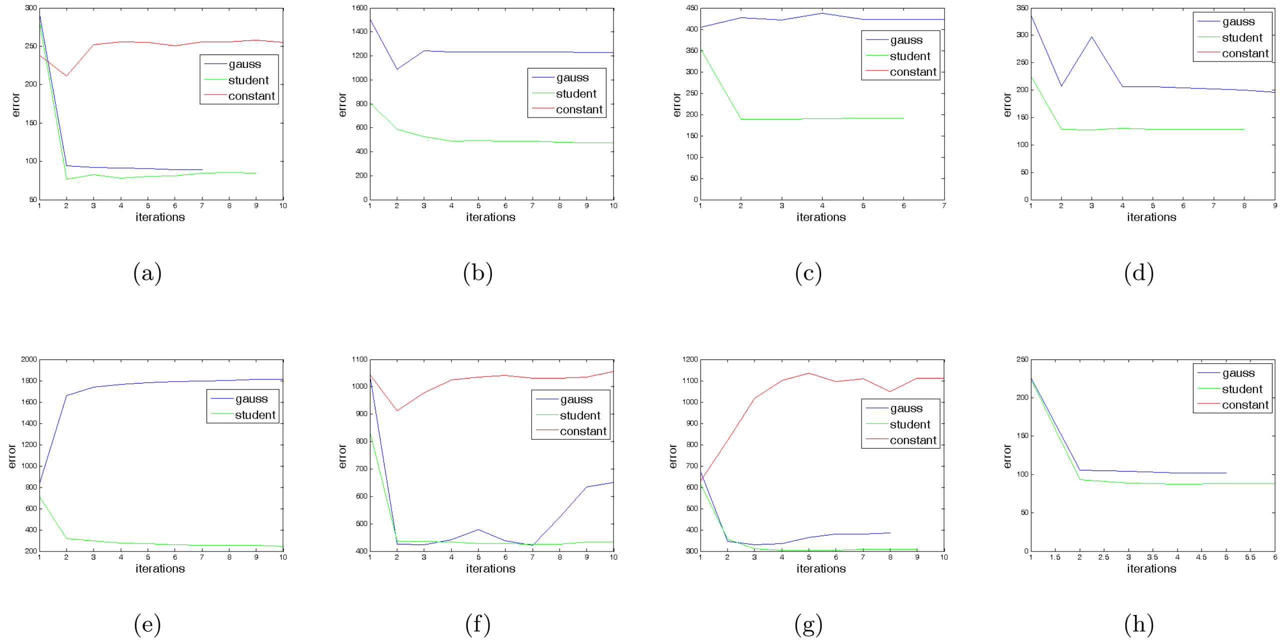


Figure 4: Typical loop cases demonstrating convergence and accuracy of the proposed Student-t based model, of the conventional Gaussian as well as the constant motion model. The constant model in several cases failed to converge. The error is calculated from (21) and the measurement units correspond to cell distance in the grid.

indicated by the following measure which had to become ideally zero:

$$d = \sum_{t=1}^{T} D(i_t, j_t) \tag{21}$$

where D is the distance-transformed image of the ground-truth path on the cell map and i_t, j_t are the coordinates of the cell where the robot is estimated to be in time t. Typical examples of convergence for loops in the dataset are presented in fig.4 after maximum ten cycles (the impression did not significantly change for more iterations). We also added a small gaussian error (μ=[-0.001, 0.001], Σ=diag(0.001,0.001)) in the odometry estimation as well as gaussian noise to the observation vectors to make the estimation problem more challenging.

We compared the approach using the Student-t mixture model for the HMM and the motion model, with the respective one using the Gaussian mixture model. We also included a method which does not adaptively learn the motion model, like in [24], however here the landmarks were computed automatically; the motion model was kept constant and was equal to the initial estimation for our adaptive model.

Clearly the Student-t based method outperformed the Gaussian-based and both outperformed the constant model. The latter gave results that were far from truth, very often worse than the initial odometry estimation and in many cases failed to converge. A typical estimation procedure for these approaches is briefly presented in Fig.5.

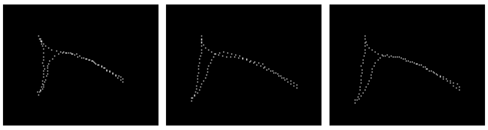
The proposed method depends on the accurate loop closure detection, however we noted that in cases of big error the estimation could not converge. Setting upper bounds to the motion model values helps eliminationg those cases. Also grid quantization errors contribute to overall error.

In our implementation, all probabilities were represented by discrete grids. To increase efficiency we have stored the motion model $p(x_{t+1}|u_t, x_t)$ in a look-up-table and we have exploited symmetry. We have also stored the $a(x_t)$, $\beta(x_t)$ into sparse arrays to save memory. The error calculation was not performed in every step but after traveling for a constant distance R=0.08 measurement units and the error as well as the observations were interpolated. The size of the grid was adapted for every loop to achieve an acceptable resolution that was much higher than R. Typical grid sizes were of 5000-20000 cells. Calculations for such sizes and T ranging from approximately 50 to 150 cycles ranged from one to ten minutes, in a standard PC using MATLAB, which can be considered acceptable for offline operation.

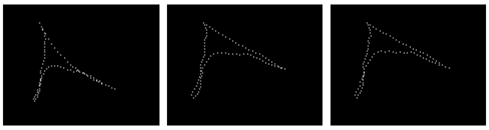
7. CONCLUSIONS

We have demonstrated the utility of incorporating a robust pdf model by applying it on a framework for detecting and exploiting loop closures to calculate a topological map.

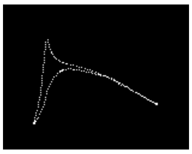
In the presented setup a HMM assisted the map creation through a SLAM approach in two main ways: in clustering the observations into landmarks and in defining what would be observed given the current position. The spatial position was calculated by employing the EM algorithm in combination with forward and backward variables. The estimator converged in reasonable time and the complexity was linear to the map size and to the number of observations.



(a)



(b)



(c)

Figure 5: Typical case of iterative trajectory calculation after loop closure detection for the case of (a) Student-t model and (b) Gaussian model. Third, fifth and tenth (final) iteration are displayed. In (c) is the ground truth.

We proposed the employment of the Student-t mixtures to represent the observation probabilities of the HMM. We saw that this choice gave higher position accuracy compared to a Gaussian mixture. This was due to outliers coming mainly from the camera blurring due to rapid motion and due to people moving in the scene during the acquisition. Such outliers were able to corrupt the Gaussian models, without however so serious effects on the respective Student-t models. We also modeled the odometry error using a mixture of the Student-t and here similar observations apply.

The presented work is expected to have impact on SLAM applications that employ HMMs to build maps in noisy environments, or which employ probabilistic models to represent the odometry error. We will investigate how similar ideas can be applied to online map estimation.

8. ACKNOWLEDGMENTS

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES

9. REFERENCES

[1] Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions. *Pattern Recognition*, 35(5):1127 – 1142, 2002.

[2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 24(5):1027–1037, 2008.

[3] N. Bellotto, K. Burn, E. Fletcher, and S. Wermter. Appearance-based localization for mobile robots using digital zoom and visual compass. *Robot. Auton. Syst.*, 56(2):143–156, 2008.

[4] N. Bellotto, K. Burn, E. Fletcher, and S. Wermter. Appearance-based localization for mobile robots using digital zoom and visual compass. *Robot. Auton. Syst.*, 56(2):143–156, 2008.

[5] K. Berns and S. Mehdi. Use of an autonomous mobile robot for elderly care. In *Advanced Technologies for Enhancing Quality of Life (AT-EQUAL), 2010*, pages 121 –126, july 2010.

[6] S. Chatzis and D. I. Kosmopoulos. A variational bayesian methodology for hidden Markov models utilizing Student's-t mixtures. *Pattern Recognition*, 44(2):295–306, 2011.

[7] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou. Robust sequential data modeling using an outlier tolerant hidden Markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1657–1669, 2009.

[8] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[9] G. Dissanayake, P. Newman, S. Clark, D. H. Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Neural Networks*, 17(3):229–241, 2001.

[10] H. F. Durrant-Whyte. Uncertain geometry in robotics. *International Journal of Robotics Research*, 4(1):23–31, Feb. 1988.

[11] A. Eliazar and R. Parr. Dp-slam: fast, robust simultaneous localization and mapping without predetermined landmarks. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1135–1142, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[12] F. Ferreira, I. Amorim, R. Rocha, and J. Dias. T-slam: Registering topological and geometric maps for robot localization in large environments. In *Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, pages 392–398, Seoul, Korea, Aug. 2008.

[13] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, May 2001.

[14] K. L. Ho and P. Newman. Detecting loop closure with scene sequences. *Int. J. Comput. Vision*, 74(3):261–286, 2007.

[15] M. Kim, S. Kim, S. Park, M.-T. Choi, M. Kim, and H. Gomaa. Service robot for the elderly. *Robotics Automation Magazine, IEEE*, 16(1):34 –45, march 2009.

[16] S. X. Liao and M. Pawlak. On the accuracy of zernike moments for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

20:1358–1364, 1998.

[17] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI*, pages 1151–1156, 2003.

[18] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257 – 286, Feb. 1989.

[19] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[20] H. Shatkay and L. P. Kaelbling. Learning geometrically-constrained hidden markov models for robot navigation: bridging the topological-geometrical gap. *J. Artif. Int. Res.*, 16(1):167–207, 2002.

[21] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[22] R. Smith, M. Self, and P. Cheeseman. *Estimating uncertain spatial relationships in robotics.* Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[23] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[24] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3-4):253–271, 1998.

[25] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal on Robotics Research*, 25(5/6):403–430, 2005.

[26] N. Tomatis. *Hybrid, Metric-Topological Representation for Localization and Mapping*, volume 38 of *Springer Tracts in Advanced Robotics*. Springer, 2008.

[27] Z. Zivkovic, O. Booij, B. Kroese, E.Topp, and H.I.Christensen. From sensors to human spatial concepts: an annotated dataset. *Transactions on Robotics*, 24(2):501–505, April 2008.

[28] Z. Zivkovic, O. Booij, and B. Kröse. From images to rooms. *Robot. Auton. Syst.*, 55(5):411–418, 2007.