Fast Re-parameterisation of Gaussian Mixture Models for Robotics Applications

Ben Upcroft, Suresh Kumar, Matthew Ridley, Lee Ling Ong, and Hugh Durrant-Whyte
Australian Centre for Field Robotics, University of Sydney, Australia
b.upcroft@cas.edu.au

Abstract

Autonomous navigation and picture compilation tasks require robust feature descriptions or models. Given the non-Gaussian nature of sensor observations, it will be shown that Gaussian mixture models provide a general probabilistic representation allowing analytical solutions to the update and prediction operations in the general Bayesian filtering problem. Each operation in the Bayesian filter for Gaussian mixture models multiplicatively increases the number of parameters in the representation leading to the need for a re-parameterisation step. A computationally efficient re-parameterisation step will be demonstrated resulting in a compact and accurate estimate of the true distribution.

1 Introduction

Many robotics applications such as tracking and data fusion need a compact but descriptive representation for communicating information either between individual autonomous agents or to a central unit. Such information could include air-borne and ground-based observations of natural features and targets from both imaging and range sensors. A common attribute of all the sensor data gathered from autonomous systems is that measurements are noisy and in many cases can only be described by general probability distributions. Thus algorithm development for robotic systems is now at the stage in which techniques for manipulating and estimating general, non-Gaussian, non-point feature information is required.

Autonomous navigation and picture compilation tasks require robust feature descriptions or models. Conventional schemes in autonomous navigation have focussed on the selection of stable point features, modelled with Gaussian noise, through the use of ranging devices (laser, sonar). While such techniques have been successfully used in autonomous air, ground and underwater vehicles, they are limited in their ability to construct accurate models of unstructured and complex environments.

Kumar *et al.* have recently shown that effective feature selection relies on the information content of the region observed by the sensor [Kumar *et al.*, 2004]. Properties in the data from a visual sensor on an autonomous ground vehicle (Fig. 1) include colour, texture, and reflectivity.

Figure 1: Incoming data from a visual sensor on an autonomous ground vehicle.

Elements of information theory quantify the information content of random variables (such as the noisy data provided by a visual sensor) to be inversely proportional to the probabilities of occurrence [Cover & Thomas, 1991]. Thus less likely states of a random variable provide greater information than more likely ones. Uniqueness in the data is therefore related to its information content which is in turn related to its frequency of occurrence. Kumar *et al.* compute the information content in an image through property histograms and address the feature selection problem by explicitly working with the least likely features.

An example of the complexity in the information content for the colour properties in the above image is graph-

ically represented in Fig. 2.

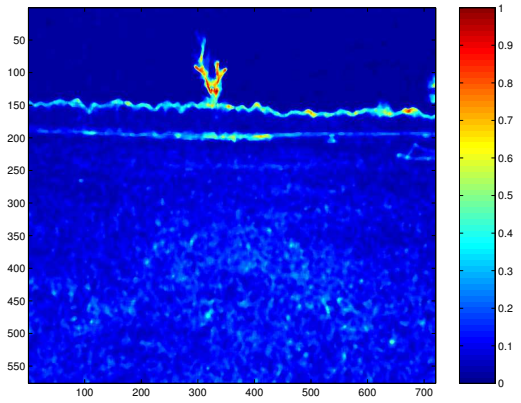

Figure 2: Information content in colour space. A two dimensional colour histogram of the raw red and green intensities in the image was generated. Subsequently, the information content of each pixel was computed using the formula $log(\frac{1}{p})$, where p is the probability of occurrence. Areas coloured red in this figure represent large amounts of information while blue small amounts.

Due to the nature of these feature properties, non-Gaussian, probabilistic representations must be considered. If the description and manipulation of non-Gaussian sensor observations are to be useful in robotics there is a need to combine general probability models in a consistent manner. Thus, it is crucial to choose a representation which offers generality but also allows computationally efficient Bayesian estimates.

Given the non-Gaussian nature of sensor observations, general probability density models for stochastic variables and the estimation of these models fall under two classes: parametric and nonparametric. Parametric density models only require a finite number of parameters to describe the distribution over the observed data with an assumption that there is some underlying structure. A Gaussian mixture model (also known as a sum of Gaussians) is a commonly used parametric distribution. Nonparametric density estimation provides a general class of methods for dealing with problems when the underlying structure of the data is poorly known. A kernel density estimator is one such method in which each data point provides evidence for a non-zero probability density *i.e.* an individual kernel function (such as a Gaussian) describes each observation.

Particle filters, a nonparametric model, are commonly used for general Bayesian methods. Grid-based methods also offer general representations but do not scale well with state dimension and thus limits their applicability. Gaussian mixture models (GMM) provide an alternative representation which allow analytical solutions to the prediction and update filtering operations that are performed. As in particle filtering, resampling [Doucet98] or, in the case of a GMM, re-parameterisation is required. Thus, for a GMM representation to be viable in the robotics and data fusion domain, re-parameterisation must be computationally fast and result in an accurate estimate of the true distribution.

The following sections illustrate the advantages of the Gaussian mixture model for Bayesian estimation methods. Section 2 introduces the form of the Gaussian mixture model. The nonlinear filtering operations using the GMM representation will be derived in Sec. 3. The particular density estimation techniques used for fast re-parameterisation will be described in Sec. 4. Sections 5 and 6 show results and future directions.

2 Gaussian Mixture Models

Multimodal densities often reflect the existence of subpopulations or clusters in the population from which the samples are taken. It is often the case that each of these subpopulations can be reasonably well modelled by a simple density such as a Gaussian. It is then possible to use a strategy in which the overall estimation problem is broken down into a set of smaller density estimation problems that have well-developed methods for obtaining accurate solutions.

A mixture density is defined for a random variable $\mathbf{X}$ as

$$P(\mathbf{x}|\theta) = \sum_{i=1}^{N} \pi_i p_i(\mathbf{x}|\theta_i) \tag{1}$$

where $\mathbf{x}$ are the observations of $\mathbf{X}$, θ_i is the parameter vector, p_i is the probability density for the ith subpopulation also known as a mixture component, and π_i are positive weights with the property $\sum_{i=1}^{N} \pi_i = 1$.

Gaussian distributions are commonly used as mixture components so that for the multi-dimensional case

$$\begin{aligned} p_i(\mathbf{x}|\theta_i) &= \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \\ &= \frac{1}{\sqrt{2\pi\Sigma_i}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)\right) \\ &= \mathcal{N}_i \end{aligned} \tag{2}$$

where μ_i is the mean and Σ_i is the full covariance.

Figure 3 illustrates an example of a Gaussian mixture model with three components. It also shows a typical sample set taken from the true distribution.

3 GMM Filter

Conventionally, the Kalman filter and the Particle filter have been used for Bayesian estimation in tracking and robotics. If nonlinear motion models are required, the Kalman filter can be replaced with the extended Kalman filter. Both the Kalman and extended Kalman

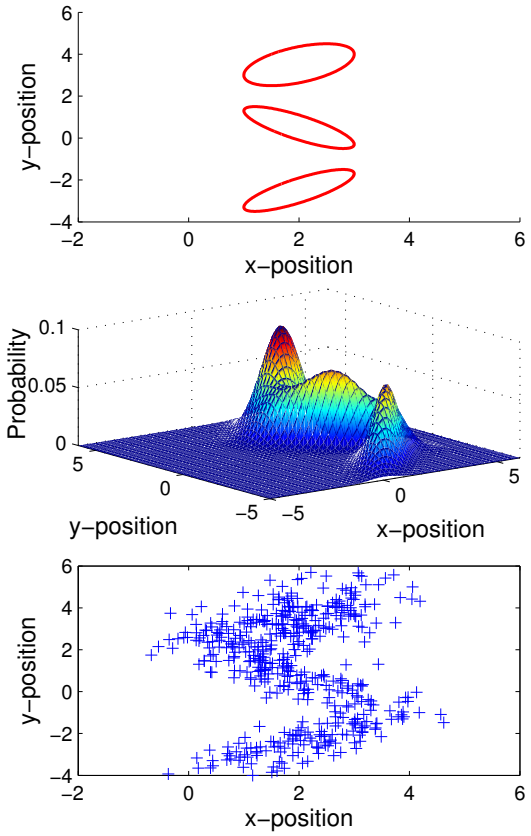

Figure 3: Mixture of three Gaussians in a two-dimensional space. Top) Contours outlining one standard deviation for each mixture component. Middle) Probability density of the mixture distribution. Bottom) 500 sample points from the distribution.

filters suffer from the restriction that the probabilistic representations must be Gaussian. However, the general Bayesian approach to filtering includes the possibility of combining non-Gaussian sensor observations in a rational manner.

Particle filters offer a general Bayesian method for combining probabilistic models. Representations of the distributions are in the form of samples. For this reason, particle filters can run into computational issues if the state space is of high dimension. An alternative general probabilistic representation is a GMM which also allows the generality of the Bayesian estimation problem to be exploited.

The following sections will show how GMMs can be incorporated in the general Bayesian estimation formulation and allow essential operations to be efficiently performed. Sec 3.1 describes Bayes theorem in which new observations in the form of GMMs are combined with prior information. Sec. 3.2 shows the derivation for the prediction of future states using GMMs.

3.1 Measurement Update: Bayes Theorem

Bayes theorem provides an incremental and recursive, probabilistic method for combining observations $\mathbf{z}_k$, of a state $\mathbf{x}_k$. A sensor observation at time t_k, is modeled as a conditional probability distribution $P(\mathbf{z}_k|\mathbf{x}_k)$. This is then incorporated with a prior belief $P(\mathbf{x}_{k-1})$ to determine a revised posterior distribution on the state:

$$P(\mathbf{x}_k|\mathbf{z}_k) = \frac{P(\mathbf{z}_k|\mathbf{x}_k)P(\hat{\mathbf{x}}_{k-1}|\mathbf{z}_{k-1})}{P(\mathbf{z}_k|\mathbf{z}_{k-1})} \tag{3}$$

where $P(\hat{\mathbf{x}}_k|\mathbf{z}_{k-1})$ is the predicted distribution calculated using the Chapman-Kolmogorov equation.

GMMs allow the update step involving Bayes theorem to be solved analytically which in general is not possible. Substitution of the general probability distributions with GMMs given in Eq.s 1 and 2 results in

$$P(\mathbf{x}_k|\mathbf{z}_k) = A\sum_{i=1}^{M}\pi_{zi}\mathcal{N}_{zi}\sum_{j=1}^{N}\pi_{xj}\mathcal{N}_{xj} \tag{4}$$

where $A = 1/P(\mathbf{z}_k|\mathbf{z}_{k-1})$ is a normalising constant, the $\mathcal{N}_z$'s are the mixture components for the likelihood distribution $P(\mathbf{z}_k|\mathbf{x}_k)$, and the $\mathcal{N}_x$'s are the mixture components for the prediction $P(\hat{\mathbf{x}}_k|\mathbf{z}_{k-1})$. Similarly for the priors π_z and π_x.

Expanding Eq. 4:

$$P(\mathbf{x}_k|\mathbf{z}_k) = A\times \begin{bmatrix} (\pi_{z1}\pi_{x1}\mathcal{N}_{z1}\mathcal{N}_{x1} & + \cdots & + \pi_{z1}\pi_{xN}\mathcal{N}_{z1}\mathcal{N}_{xN}) \\ + (\pi_{z2}\pi_{x1}\mathcal{N}_{z2}\mathcal{N}_{x1} & + \cdots & + \pi_{z2}\pi_{xN}\mathcal{N}_{z2}\mathcal{N}_{xN}) \\ & \vdots & \\ + (\pi_{zM}\pi_{x1}\mathcal{N}_{zM}\mathcal{N}_{x1} & + \cdots & + \pi_{zM}\pi_{xN}\mathcal{N}_{zM}\mathcal{N}_{xN}) \end{bmatrix} \tag{5}$$

So each term reduces to a multiplication of two weighted Gaussians with an analytical solution:

$$\mathcal{N}_z\mathcal{N}_x = k_{zx}\frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_{zx}|^{1/2}}e^{-\frac{1}{2}[\mathbf{x}-\mu_{zx}]^T\mathbf{\Sigma}_{zx}{}^{-1}[\mathbf{x}-\mu_{zx}]} \tag{6}$$

where n is the dimensionality of the state,

$$k_{zx} = \frac{|\mathbf{\Sigma}_{zx}|^{1/2}}{(2\pi)^{n/2}|\mathbf{\Sigma}_z|^{1/2}|\mathbf{\Sigma}_x|^{1/2}} \tag{7}$$

$$\times\, e^{-\frac{1}{2}\left[\mu_z^T\mathbf{\Sigma}_x{}^{-1}\mu_z+\mu_x^T\mathbf{\Sigma}_x{}^{-1}\mu_x-\mu_{zx}^T\mathbf{\Sigma}_{zx}{}^{-1}\mu_{zx}\right]} \tag{8}$$

with

$$\mathbf{\Sigma}_{zx}{}^{-1} = \mathbf{\Sigma}_z{}^{-1} + \mathbf{\Sigma}_x{}^{-1} \tag{9}$$

and

$$\mu_{zx} = \mathbf{\Sigma}_{zx}\left(\mathbf{\Sigma}_z{}^{-1}\mu_z + \mathbf{\Sigma}_x{}^{-1}\mu_x\right) \tag{10}$$

Since an analytical solution can be obtained, numerical computation is very fast. However, each update increases the number of Gaussian mixture components resulting in the need for a step in which a reduction of the parameter set must occur.

3.2 Prediction: The Chapman-Kolmogorov Equation

The prediction step is performed in between observations and is achieved using the Chapman-Kolmogorov equation:

$$P(\hat{\mathbf{x}}_k|\mathbf{z}_{k-1}) = \int P(\mathbf{x}_k|\mathbf{x}_{k-1})P(\mathbf{x}_{k-1}|\mathbf{z}_{k-1},\mathbf{x}_0)d\mathbf{x}_{k-1} \tag{11}$$

where the transition probability density $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is known as the motion model, $P(\mathbf{x}_{k-1}|\mathbf{z}_{k-1},\mathbf{x}_0)$ is the updated estimate from the previous time step, and x_0 is the initial state.

The Chapman-Kolmogorov equation involves the knowledge of the state $\mathbf{x}_{k-1}$ at time t_{k-1} summarised by the probability distribution $P(\mathbf{x}_{k-1})$. The motion model $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ describes the stochastic transition from a state $\mathbf{x}_{k-1}$ at time t_{k-1} to a state $\mathbf{x}_k$ at time t_k. This transition is related to an underlying model of the target given by $\mathbf{x}_k = \mathbf{f}(\mathbf{x_{k-1}}, \mathbf{u_k})$. Thus, the Chapman-Kolmogorov equation is the summation of the product between the probability of the target being at a particular state $\mathbf{x}_{k-1}$ and the probability of it evolving to all other states $\mathbf{x}_k$, over all possible states $\mathbf{x}_{k-1}$. This evolution equation is therefore a convolution between the motion model and the prior distribution.

As with Bayes theorem, GMMs allow an analytical solution for the Chapman-Kolmogorov equation. Again this ensures real time numerical computation in the update and prediction steps. Substituting GMMs into Eq. 11 for the two distributions results in a similar equation to Eq. (5) but with each term being a convolution between two weighted Gaussians rather than a multiplication.

It can be shown that a convolution between two Gaussians is also a Gaussian with the form [Norwich, 2003]

$$\pi\mathcal{N}\left(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2\right) \tag{12}$$

where the subscripts denote the variables for the two Gaussians and π is a constant weighting term.

Similarly to the Bayes update step, the solution to the Chapman-Kolmogorov equation is a weighted sum of $M \times N$ Gaussians. It must be noted that due to the multiplicative increase in this prediction step a re-estimation step is also needed here if the number of Gaussian mixture components is to remain small.

4 Density Estimation for GMMs

As demonstrated in the previous section, both the update and prediction for GMMs result in a multiplicative increase of the mixture components. Thus, similarly to the resampling step for a particle filter, a re-parameterisation or re-estimation step is required.

The expectation-maximisation (EM) algorithm provides a general approach to the problem of maximum likelihood (ML) parameter estimation in statistical models with variables that are not observed. An example of such hidden variables are the underlying mixture components in a GMM. It is possible to use a numerical optimisation routine for maximising the likelihood and subsequently estimating the parameters. However, it would be useful to take advantage of the underlying structure of the model, breaking the optimisation problem into manageable pieces. EM ensures that this is implemented in a systematic way.

EM is an iterative algorithm involving two steps. For a GMM, the E-step involves calculating the probability of the ith mixture component given the data and parameters

$$p(h_n^i|x_n,\mu_i,\Sigma_i) = \frac{p(x_n|h_n^i,\mu_i,\Sigma_i)p(h_n^i|\pi_i)}{p(x_n|\mu_i,\Sigma_i)} \tag{13}$$

$$= \tau_n^i \tag{14}$$

where h^i is an unobserved variable equal to 1 if the data point x_n, belongs to the ith mixture component and 0 otherwise. Note that the E-step is a form of Bayes theorem and for this reason τ^i is referred to as the posterior and π_i as the prior.

The M-step involves calculating the parameters; the weighted sample means, covariances, and priors, which are used as inputs in the next iteration of the E-step.

$$\mu_i = \frac{\sum_{n=1}^N \tau_n^i x_n}{\sum_{n=1}^N \tau_n^i} \tag{15}$$

$$\Sigma_i = \frac{\sum_{n=1}^N \tau_n^i (x_n-\mu_i)^T(x_n-\mu_i)}{\sum_{n=1}^N \tau_n^i} \tag{16}$$

$$\pi_i = \frac{1}{N}\sum_{n=1}^N \tau_n^i \tag{17}$$

Implementation of these two steps provides a simple algorithm which converges to the maximum likelihood. However, EM is sensitive to parameter initialisation and can converge to a local maximum rather than the true value for the maximum likelihood. Convergence can also be very slow if the initial parameters are particularly bad compared to the true values.

The computational complexity of the EM algorithm for GMMs is $O(i \times ND^2)$ where i is the number of iterations performed, N is the number of samples, and D is the dimensionality of the state.

To ensure quick convergence, a number of other algorithms can be used for initialisation, of which k-means clustering is one [Mackay03]. The k-means algorithm tries to assign the nearest cluster mean (or centre) to each data point. Once the assignments are known, a new cluster mean can be calculated using a sampled average.

Unfortunately k-means suffers from poor computational scalability for large numbers of multi-dimensional data points. Pelleg and Moore have addressed this issue of speed using the X-means algorithm which embeds the dataset into a multiresolutional kd-tree and stores sufficient statistics at its nodes [Pelleg & Moore, 2000]. The computational complexity for a naive k-means algorithm is $O(kDN)$ where k is the number of clusters. With the acceleration modifications in X-means, the complexity reduces to $O(D)$ [Pelleg & Moore, 1999].

Additionally, X-means efficiently determines the number of clusters by continuously splitting existing cluster centres and comparing k-means results using the Bayesian Information Criterion. Most searches for the number of clusters proceed sequentially, trying each value of k. This results in a computational complexity of $O(k_{max})$. X-means improves the structure on which to search decreasing the complexity to $O(\log k_{max})$.

This fast implementation of k-means gives a reasonable parameter initialisation for the EM algorithm ensuring that only a few iterations are needed before convergence to the ML is achieved.

5 Results

This section illustrates the effectiveness of the density estimation techniques presented above. The estimated distribution will be visually compared with the true distribution and the computational time will also be given.

5.1 Comparison of the Estimated and True Distribution

It is often the case that density estimation techniques are evaluated with test distributions that have well separated modes. However, the multimodal distributions that result from sensor observations are commonly close together, non-spherical, and of different sizes. A typical test distribution used to evaluate the combination of X-means and EM is shown in Fig. 4.

Sampling from known Gaussian mixture components is very efficient. 5000 samples were taken from the true distribution and used for the density estimation methods. In two dimensions, X-means consistently resulted in good parameter initialisation with a sufficient number of data points. In the case of mixtures with less than five components, 5000 data points were required. Further studies are needed that address the scalability of this algorithm with comparisons to resampling for a particle filter. Figure 5 explicitly illustrates the difficulty in determining the separate modes when only the samples are observed.

Figure 6 compares the true probability distribution (top), the estimated distribution after X-means using 5000 samples from the true distribution (middle), and the estimated distribution once the EM algorithm has been performed using the parameters from X-means as initial conditions (bottom). Note that the X-means estimate is quite poor due to the implicit assumption that the underlying mixture components are spherical Gaussians. This incorrect assumption compounded with the small separation of modes in the distribution can result in faulty determination of centres. However, X-means provides reasonable initialisation parameters for the EM algorithm.

Figure 7 shows the contours at one standard deviation of the original distribution. Also shown are the resulting position and number of centres from the X-means algorithm and the contours from EM. As can be seen, X-means has performed quite well and has found positions for the centres which are close to the true values. The EM algorithm is able to use these centres and within 20 iterations can find a good estimate of the true distribution.

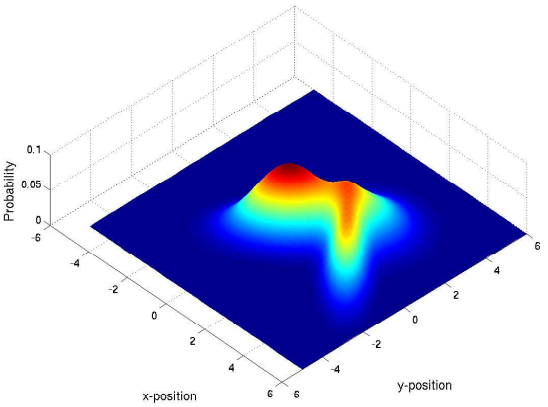

Figure 4: One of the typical test distributions used for evaluation of the described density estimation techniques. Note that the modes are closely spaced and of different size.

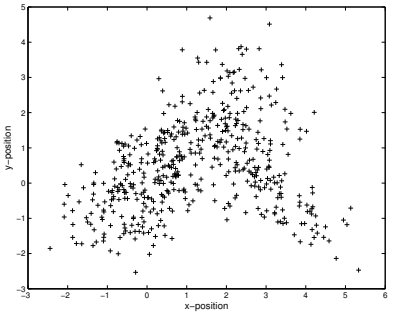

Figure 5: 500 samples from the mixture distribution in Fig. 4. Note that it is quite hard to determine the three modes.

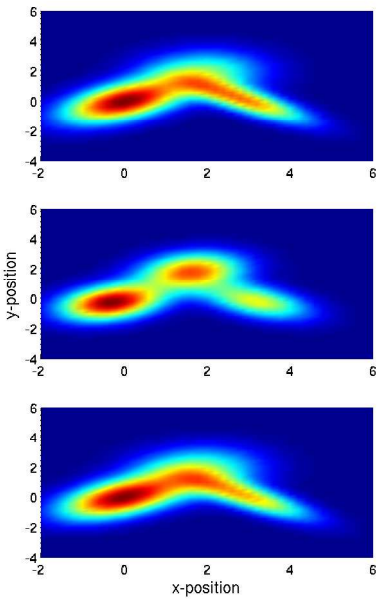

Figure 6: Comparison between the true and estimated distribution after X-means and EM. Top) The true probability distribution. Middle) The estimated distribution after X-means. Note that the estimate is poor as X-means assumes that the Gaussian mixture components are spherical. Bottom) Final estimate after EM. This distribution matches quite closely to the original.

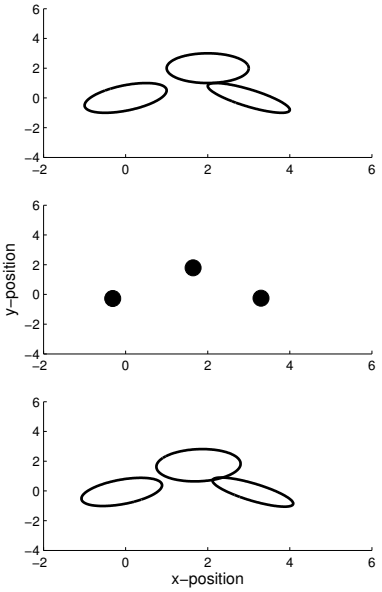

Figure 7: Comparison between the true and estimated distribution after X-means and EM. Top) Contours at one standard deviation of the distribution in Fig 4. Middle) Centres found by X-means. Bottom) Estimated covariances after EM.

5.2 Computational Speed

Code for the X-means algorithm was provided by Pelleg and Moore and is implemented in C [Pelleg & Moore, 2004]. The determination of the number and position of centres by X-means takes 0.31s on average for 5000 data points using a 1.5GHz Pentium Centrino.

Since X-means provides reasonable initialisation paramaters, only a small number of EM iterations need to be performed. Thus computational time remains minimal, although EM is relatively computationally slow. Estimation using a Matlab implementation of EM on the same computer as above takes 0.57s. Significant gains in speed could be achieved if the EM algorithm was implemented in a language such as C/C++. Additional performance enhancements could be achieved using approximations to the EM algorithm such as Variational Bayesian EM [Beal & Ghaheramani, 2003]. This particular method allows much faster convergence and also allows the use of smaller data sets, again decreasing computational time.

6 Conclusion

This paper has shown that Gaussian mixture models can be used as compact representations for complex models of the world. In addition, analytical solutions were derived for the general Bayesian filtering problem often used in robotics and tracking. The major hurdle preventing GMMs from being a computationally efficient tool for the autonomous systems domain is the multiplicative increase of parameters in each of the Bayesian filtering operations.

A solution to this problem was presented here in the form of a re-parameterisation step. This re-estimation involved a computationally fast clustering algorithm, X-means, which was used to initialise parameters for the EM algorithm. Only a few iterations of EM were subsequently needed to ensure an accurate estimate of the true distribution. Since this density estimation step was shown to be computationally fast, it potentially can be performed after each Bayesian observation update for many existing sensor data rates. Thus, GMMs are a viable representation for probabilistic modelling and Bayesian filtering.

A number of improvements to these estimation algorithms are currently being investigated. Firstly, the statistical measure for determination of the number of modes in X-means can be significantly improved. At present, there is an assumption that the mixture components are spherical. Incorporation of a statistical measure which allows for non-spherical Gaussian mixture components is needed if accurate determination of the number and position of modes is required. This should also allow a smaller number of data points to be needed without loss in the quality of parameter initialisation.

Additionally, approximations to the EM algorithm, such as the Variational Bayesian EM algorithm, can also provide a significant decrease in computational time. With these improvements, it is envisioned that real-time performance for combining all incoming sensor data, modelled as GMMs, will indeed be possible.

References

[Beal & Ghaheramani, 2003] M. Beal and Z. Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data: with Application to Scoring Graphical Model Structures. *Bayesian Statistics*, Vol. 7, Oxford University Press, 2003

[Cover & Thomas, 1991] T. Cover and J. Thomas. Elements of Information Theory. J. Wiley and Sons, 1991

[Doucet, 1998] A. Doucet. On Sequential Simulation-Based Methods for Bayesian Filtering. *Technical Report*, Signal Processing Group, Dept. of Engineering, University of Cambridge, 1998.

[Kumar *et al.*, 2004] S. Kumar and S. Scheding. *Probabilistic Feature Extraction.* to be submitted.

[MacKay, 2003] D. MacKay Information Theory, Inference, and Learning Algorithms. http://www.inference.phy.cam.ac.uk/mackay/itila/ 2003

[Norwich, 2003] K. Norwich Information, Sensation, and Perception. http://www.biopsychology.org/norwich/isp/isp.htm 2003

[Pelleg & Moore, 1999] D. Pelleg and A. Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. In *Proceedings of the 5th International Conference on Knowledge Discovery in Databases*, pages 277, 1999.

[Pelleg & Moore, 2000] D. Pelleg and A. Moore. X-means: Extending K-means With Efficient Estimation of the Number of Clusters. In *Proceedings of the 17th International Conference on Machine Learning*, pages 727, 2000.

[Pelleg & Moore, 2004] D. Pelleg and A. Moore. http://www-2.cs.cmu.edu/ dpelleg/kmeans.html