Mutual Information Based Data Selection in Gaussian Processes for People Tracking

Zulkarnain Zainudin
Faculty of Electronics and Computer Engineering
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, MALAYSIA
Email: zul@utem.edu.my

Sarath Kodagoda and Linh Van Nguyen
Centre for Autonomous Systems (CAS)
Faculty of Engineering and Information Technology
University of Technology, Sydney
PO Box 123 Broadway, Ultimo, NSW 2007, Australia
Email: {Sarath.Kodagoda@, van.l.nguyen@student.}uts.edu.au

Abstract

It is the general perception that models describing human motion patterns enhance tracking even with long term occlusions. One effective way of learning such patterns is to use Gaussian Processes (GP). However, with the increase of the amount of training data with time, the GP becomes computationally intractable. In this work, we have proposed a Mutual Information (MI) based technique along with the Mahalanobis Distance (MD) measure to keep the most informative data while discarding the least informative data. The algorithm is tested with data collected in an office environment with a Segway robot equipped with a laser range finder. It leads to more than 80% data reduction while keeping the rms errors within the required bounds. We have also implemented a GP based Particle filter tracker for long term people tracking with occlusions. The comparison results with Extended Kalman Filter based tracker shows the superiority of the proposed approach.

1 Introduction

Tracking people are important in security, surveillance and human robot interaction. There have been much research and interest in people tracking in populated environment using sensors such as cameras and laser range finders (LRFs) with various detection and tracking techniques. Heart of the people tracking techniques consists of two important aspects; state estimation and data association. In various applications, the state estimation has been carried out using variety of algorithms such as Kalman Filter (KF) [A.Fod et al.,2002] and Particle Filter (PF) [P.Chakravarty et al., 2005]. Further, there are other concepts such as the interactive multiple model (IMM) [Zainudin et al., 2010a] has been proposed to improve tracking performances, however it still incapable of handling highly maneouvering targets and long term occlusions.

One way to resolve such problems is to learn the target behaviours and use them in the prediction stage. It is an observation that people tend to follow general patterns in various environments due to environmental and physical motion constraints. Those motion patterns are very informative and can be learned. In this direction, J. Ko et al [J.Ko et al., 2008] have proposed Gaussian processes to learn motion patterns and integrated the learned models in a particle filter tracker (GP-PF). Gaussian Processes (GPs) learn probabilistic regression models from training data by combining extreme modelling flexibilities with consistent uncertainty estimates. These probabilistic models can be readily incorporated with probabilistic filtering techniques such as Kalman filter, unscented Kalman filter and particle filter [Plagemann et al., 2007].

As the increase of data can lead to computational intractability in GP learning, the work reported in [J.Ko et al., 2010b] has long terms problems. It also can easily lead to instabilities due to insignificant special variations in data. We argue that it is not always required to keep all the data which do not provide additional information. Therefore, we have propose an approach to keep only the maximally informative data points using Mutual Information [C.Guestrin et al., 2005b] and Mahalanobis Distance (MI) criteria. .

This paper is organized as follows. In section 2, we present the details of Gaussian Process. Section 3 presents training data management. Section 4 presents the experimental setup. Section 5 discusses experimental results with three processes of optimized number of points. Section 6 concludes the paper with direction of future research.

2 Gaussian Process

A Gaussian Process is a powerful non-parametric tool for learning regression functions from sample data which contains of a collection of random variables with any subset of them having a joint Gaussian distribution [Rasmussen et al, 2005a]. It represents posterior distributions over functions based on training data. Let us assume a set of training data, $D = \langle X, \mathbf{y} \rangle$, where

$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a matrix containing d-dimensional input examples and $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ is a vector containing scalar training output.

Gaussian Processes assumes that the data is drawn from a noisy process. Its regression output is modeled using a noisy version of function,

$$\mathbf{y} = f(\mathbf{x}) + \epsilon \ , \tag{1}$$

where ϵ is zero mean additive Gaussian noise with a variance of σ_n^2. With training data, $D = \langle X, \mathbf{y} \rangle$ and a test input, $\mathbf{x}_*$, a GP defines a Gaussian predictive distribution over the output $\mathbf{y}_*$ with mean

$$\mathrm{GP}_{\mu}(\mathbf{x}_*, D) = \mathbf{k}_*^T [K + \sigma_n^2 I]^{-1} \mathbf{y} \tag{2}$$

and variance

$$\mathrm{GP}_{\Sigma}(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T [K + \sigma_n^2 I]^{-1} \mathbf{k}_* \tag{3}$$

$\mathbf{k}_*$ is a vector defined by kernel values between the test input $\mathbf{x}_*$ and the training inputs $\mathbf{x}$. K is the $n \times n$ kernel matrices of training input values where $\mathbf{k}[m] = k(\mathbf{x}_*, \mathbf{x}_m)$ and $K[m,n] = k(\mathbf{x}_m, \mathbf{x}_n)$. The variance GP_{Σ}, which is the prediction uncertainty, depends on the process noise and the correlation between the test input and the training data. The widely used kernel function, squared exponential is chosen for this application, which is given by,

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}')W(\mathbf{x}-\mathbf{x}')^T} \tag{4}$$

where σ_f^2 is the signal variance. The diagonal matrix W contains the length scales for each input dimension.

The GP parameters describing the kernel function and the process noise respectively called the hyperparameters of the Gaussian process. These hyperparameters can be learned by maximizing the log likelihood of the training data using numerical optimization techniques such as conjugate gradient descent [Rasmussen et al, 2005a]. Consider a d-dimensional trajectory V that has $|V|$ points. If we observe a set of points, $A \subset V$, based on GP model we can predict value at any point $y \in V \setminus A$. Let us denote Z_A as a set of values at the finite set A, and z_y as a value at y. In probabilistic terms, we derive the conditional distribution at predicted point of y, given Z_A as follows:

$$\mu_{y|A} = \mu_y + \Sigma_{yA} \Sigma_{AA}^{-1} (Z_A - \mu_A) \tag{5}$$

$$\sigma_{y|A}^2 = k(y,y) - \Sigma_{yA} \Sigma_{AA}^{-1} \Sigma_{Ay} \tag{6}$$

where Σ_{yA} is a covariance vector with one entry for each $\mathbf{x} \in A$ with value $k(y, \mathbf{x})$; $\mu_{y|A}$ ($\sigma_{y|A}^2$) are conditional mean and variance at y; μ_A is a mean vector of Z_A ; and Σ_{AA}^{-1} is a covariance matrix of Z_A, with every entry calculated by $k(\mathbf{x},\mathbf{x})$.

3 Training Data Management

Usual practice is to incorporate all the samples in the training phase of the GP. However, this becomes inefficient if the GP needs to be trained to accommodate new observations. In the people tracking scenarios presented in this paper, the motion models are learned and it needs to be adapted in time to accommodate variations. Therefore, with every new measurement the GP needs to be learned with increasing number of samples. Intuitively, the new observations need to be only included, if they are informative. This problem is proposed to resolve using a Mutual Information (MI) based strategy and Mahalanobis Distance (MD) based criteria. The MI picks the most informative measurements given the whole scan and uses it to represent the GP surface. Whenever new data is available MD is calculated between the new measurement and the GP. If it is within the 95% probability interval, the new measurement is discarded as the GP is already capable of representing the data. However, if the MD is larger than the 95% probability interval, the data is not representative of the GP and hence needs to be included to represent the new GP. This process will govern the data management and adapting to new scenarios.

3.1 Mutual Information

As discussed above, the most informative data points are selected based on Mutual Information algorithm. [C.Guestrin et al., 2005b].

In order to find *k* best points in whole trajectory V, it starts from empty set of locations $A=\phi$ and greedily add placements in sequence until $|A|=k$. The algorithm chooses the next point that produces the maximum increase of mutual information. More specifically, the MI between the subset *A* and the rest of trajectory $V\backslash A$ can be formulated by:

$$F(A)=I(A;V\backslash A)$$

Once $y \in V \setminus A$ is chosen and added to *A*, the variation of MI can be calculated by:

$$\begin{aligned} &F(A \cup y) - F(A) = \\ &= H(A \cup y) - H(A \cup y|\bar{A}) - [H(A) - H(A|\bar{A} \cup y)], \\ &= H(y|A) - H(y|\bar{A}), \end{aligned} \tag{7}$$

where $A \cup y$ denotes the $A \cup \{y\}$ and $\bar{A}$ represents $V\backslash(A \cup y)$; $H(y|A)$ is the conditional entropy of y after observing A. By maximizing $F(A \cup y) - F(A)$, we will find the next point y to add into A. Suppose that the values in V are Gaussian random variables, then (7) can be calculated in detail by:

$$F(A \cup y) - F(A) = \frac{1}{2}\left[\log(\sigma_{y|A}^2) - \log(\sigma_{y|\bar{A}}^2)\right]$$

3.2 Mahalanobis Distance

The MD is used to decide the importance of new data to be incorporated in the GP learned model. As discussed before, this will allow the GP to represent dynamically changing environments and hence improve the

adaptability. Let's assume a new measurement value of mean μ_{xm} and variance σ_{xm} at a location $\mathbf{x}_i$, where $\mathbf{x} = \langle x, y \rangle$, was received. Now the GP can be used to predict the mean μ_{xp} and variance σ_{xp} at that location as well. Thus the MD can be calculated as

$$d(\mathbf{x}) = \sqrt{\frac{(\mu_{xm} - \mu_{xp})^2}{\sigma_{xm}^2 + \sigma_{xp}^2}} \quad . \tag{8}$$

The measurement used in this particular application is one dimensional and therefore, using chi-square tables the threshold for $d(\mathbf{x})$ to be within 95% confidence interval can be chosen as 3.84 [G.Shay, 2007].

3.3 Gaussian Process – Particle Filter (GP-PF)

Particle filter is an approximation method for nonlinear dynamic sequential modeling. Gaussian process regression can be directly applied to the problem of learning prediction and observation models required by particle filters. In this work, we restrict ourselves to train prediction models as at a later stage the observer is dynamic making is difficult to learn observation models. The prediction model maps the state and control, $(\mathbf{x}_k, \mathbf{u}_k)$ to the state transition, $\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. Then the next state can be found by simply adding the state transitions to the previous state. Therefore, appropriate form of prediction and observation training data sets is given by,

$$D_p = \langle (X, U), X' \rangle \tag{9}$$

where X is a matrix containing the locations and $X' = [\Delta x_1, \Delta x_2, \cdots, \Delta x_k]$ is a matrix containing transitions made from those states when applying the controls stored in U.

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \approx N\left(\mathrm{GP}_\mu\left([\mathrm{x}_{k-1}, \mathbf{u}_{k-1}], D_p\right), GP_\Sigma\left([\mathrm{x}_{k-1}, \mathbf{u}_{k-1}], D_p\right)\right) \tag{10}$$

Basic task of particle filter is to represent posteriors over the state $\mathbf{x}_k$ by setting X_k of weighted samples:

$$X_k = \left\{ \langle \mathbf{x}_k^m, w_k^{(m)} \rangle \middle| m = 1, \cdots, M \right\} \tag{11}$$

Here each $\mathbf{x}_k^m$ is a sample and each $w_k^{(m)}$ is a non-negative numerical factor called importance weight. Thus, GP $\left([\mathrm{x}_{k-1}, \mathbf{u}_{k-1}], D_p\right)$ is the short form of the Gaussian represented by $\left(\mathrm{GP}_\mu\left([\mathrm{x}_{k-1}, \mathbf{u}_{k-1}], D_p\right), GP_\Sigma\left([\mathrm{x}_{k-1}, \mathbf{u}_{k-1}], D_p\right)\right)$.

Note that the covariance of this prediction is typically different for each sample, taking the local density of training data into account. The complete step can be found in [J.Ko et al., 2008].

4 Experiments

The robot used in our experiments is a Segway equipped with sensors and computers. It has an onboard computer, AMD Athlon II X2 255/ Dual Core/ 3.1 GHz with 4GB DDR3 running on Linux Ubuntu 9.10 operating system. This system uses HOKUYO UTM-30LX laser range finder that has 30 meters of detection range, 0.25° angular resolution, 270° of angular field of view and 25milliseconds sampling period. The Segway robot is used in a stationary position to monitor the environment. The experiments were carried out in a common area of our faculty as shown in Fig.1.

Figure 1: Common area of our faculty.

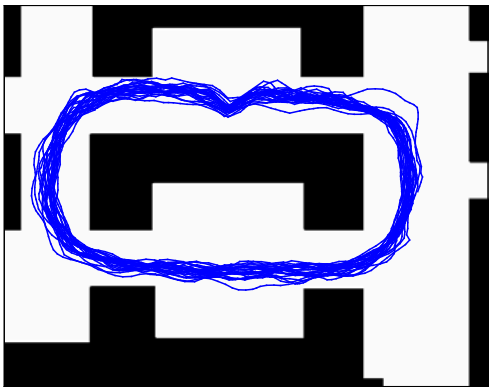

Figure 2: Multiple trajectories of a walking person.

In this experiment, a subject was walking around cubicles multiple times while the robot was observing the subject. A trace of such a trajectory containing 20 loops is shown in Fig.2.

Our method of detecting people is based on laser data that were taken at torso height of human which comprises of extracting significant features followed by a classification process [Zainudin et al., 2010a]. Once a person was detected based on the laser data, they were used for modeling the GP. The idea was to keep the root mean square error of the predictions below 5cm as the environment contains corridors of width ranging from 130 ~150cm. Then the GP-PF is used for people tracking.

5 Results

The first part of the analysis is designed to show the procedure involved in the proposed algorithm and to discuss the computational savings.

In this experiment, the training data has been processed in three stages: Firstly, applying MI on the first set of data in the trajectory to determine the least number of points that can be used to represent the GP with the given rms errors. Secondly, the MD is applied to each new measurement to ensure its contribution to new knowledge. Thirdly, we apply the MI again to remove any redundant data points with the addition of new data.

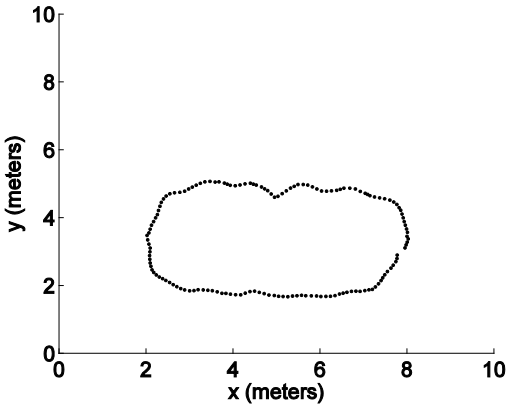

Figure 3: Points in trajectory before MI

The Fig.3 contains 150 points collected on a single trajectory. The learned GP means and covariances are shown in Fig. 5 and Fig. 6. As expected, it can be noted that the GP predictions are highly uncertain towards the corners of the plots where no measurements can be found. The data was then processed using the MI based data selection to achieve the set of points which represent the same rms error. The resulting points are shown in Fig. 4. It leads to 60% reduction of data.

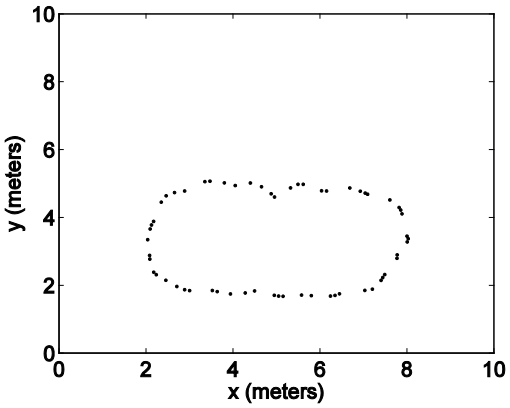

Figure 4: Points in trajectory after MI

The rms errors calculated at measured points with optimized GP is shown in Fig.7. The rms error is less than 0.045 meter.

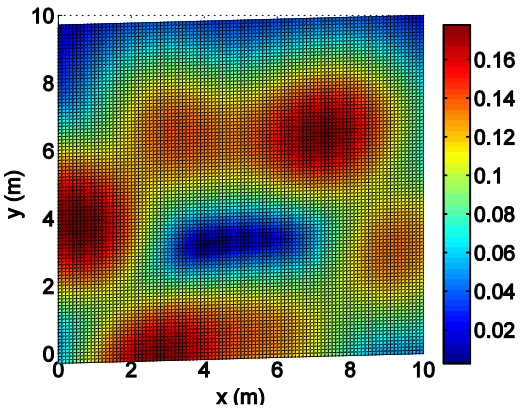

Figure 5: GP regression: Mean Values with 150 data points

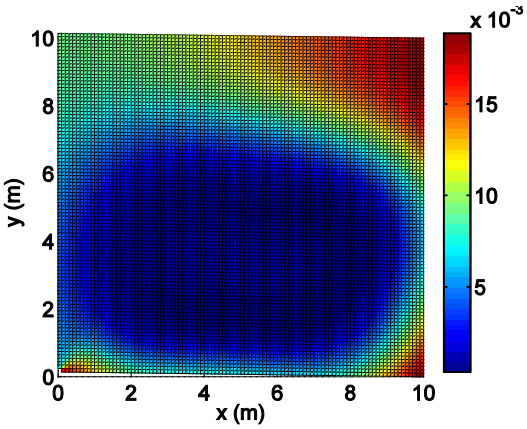

Figure 6: GP regression: Covariance Values with 150 data points

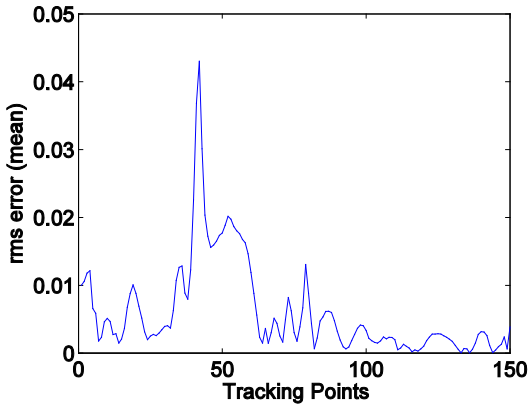

Figure 7: RMS Error between Predicted Mean and the measurements

When a new trajectory is available for the training data, the predicted and measured values of mean and covariance of each point in x and y axes will be compared using MD. Those points which have MDs less than 3.84 (threshold) are discarded. For example, most of the new measurements shown in Fig.8 provide less MDs than the threshold obeying the learned model and hence measurements that less than threshold are discarded. However, with more variations of data as can be seen in

Fig. 9 give rise to higher MDs than the threshold. Those are incorporated to the training samples for the retraining purposes.

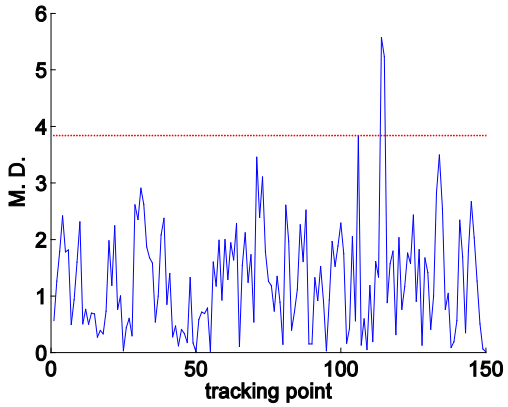

Figure 8: Mahalanobis Distance in the x direction

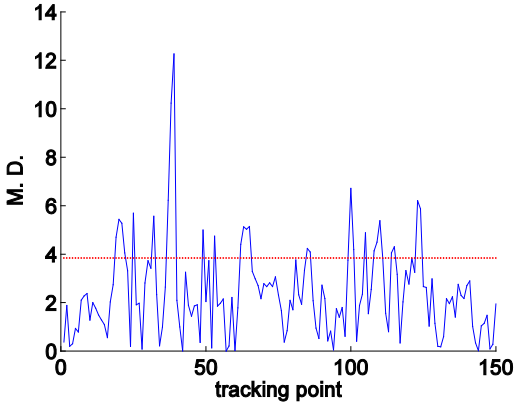

Figure 9: Mahalanobis Distance in the y direction

After the second set of observations, once the data to be added to the model has been decided based on MD, MI is used for selecting most informative data points as shown in Fig.10. Fig.11 and Fig.12 shows the mean and covariance after retraining the data still having the highest rms error of 0.014 meter as referred to Fig.13.

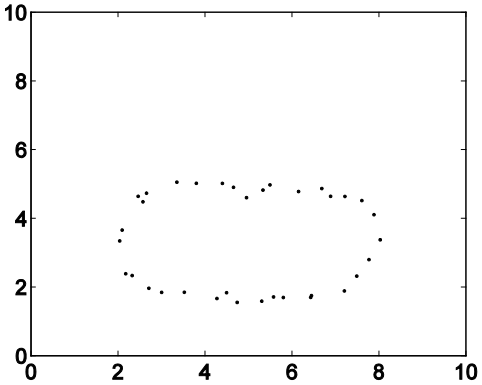

Figure 10: Points in trajectory after MD and MI

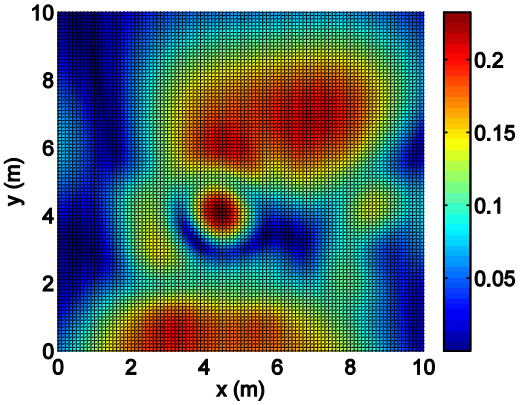

Figure 11: GP Regression: Mean Values after MD and before MI

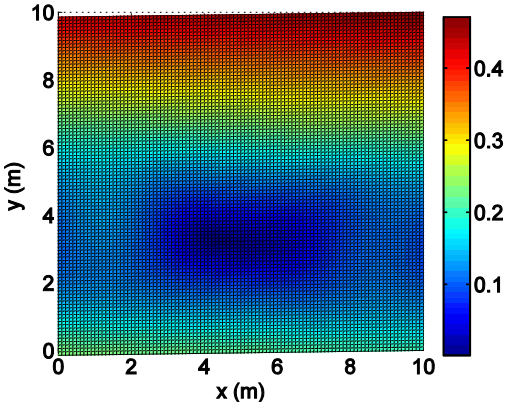

Figure 12: GP Regression: Mean Values after MD and after MI

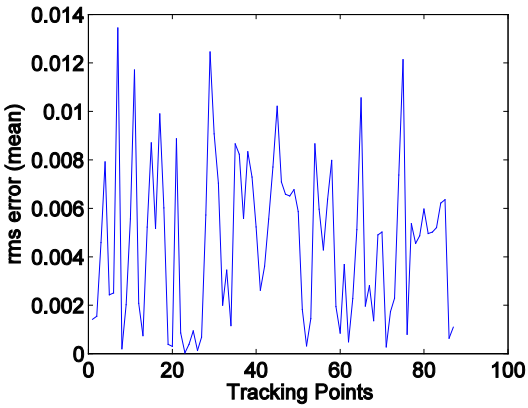

Figure 13: RMS Error between Predicted Mean

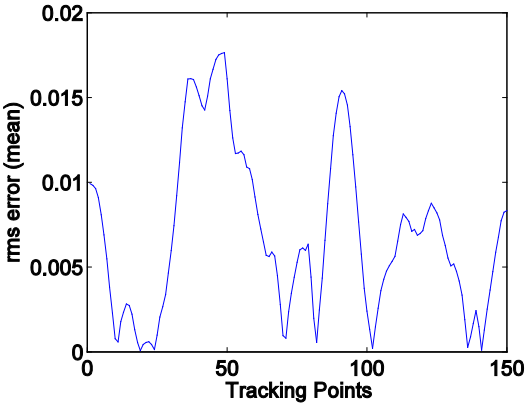

Figure 14: RMS Error between the initial and final GP

For testing the overall accuracy, referring to Fig.5 as initial training data and Fig.12 as final training data, we provide rms error of mean value as shown in Fig.14, at each point of 150 training data. The number of training data is reduced to 35 points from the total of 300 points, which is an 88% reduction of data.

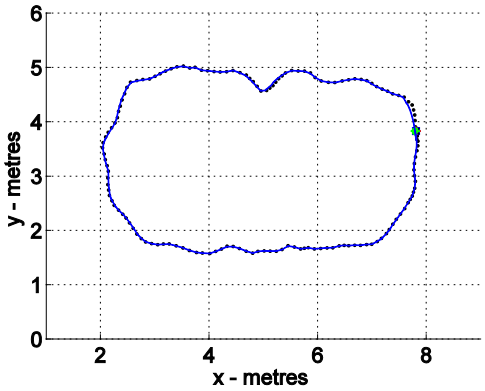

Figure 15: Tracking with GP-PF with tracking data.

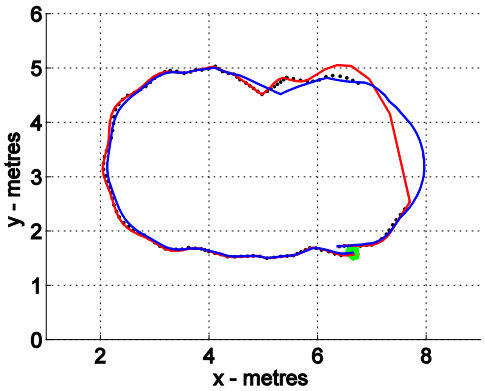

Figure 16: Tracking with GP-PF(blue line) and EKF (red line) with an occlusion.

The GP learned model is then used for predicting the PF based tracker for evaluating the long term tracking ability. Fig.16 shows the comparison results of model based EKF and the proposed GP-PF with an occlusion. As can be seen the EKF has poor tracking performance while the GP-PF still managed to perform well. It is to be noted that the EKF tracker has lost the track, but with growing covariance it could finally converge to the track again. This is possible as the experiment only involved one track (data association is assumed) and the fact that it goes around loops.

6 Conclusion and Future Works

We have implemented a MI and MD based criteria for discarding least informative data points while efficiently training a GP model. The proposed algorithm produced very appealing 88% of data reduction while keeping the rms error within bounds. This will give rise to computational savings specially when working with an extended period of time. The learned GP was then incorporated in the PF for people tracking. It has produced better tracking performance with occlusions when comparing with EKF.

In our future work, we intend to test this algorithm in more complex scenarios with junctions, occlusions and multiple people.

References

[Zainudin et al., 2010a] Z.Zainudin, K.R.S. Kodagoda, G. Dissanayake. Torso Detection and Tracking using a 2D Laser Range Finder. *Proceedings of 2010 Australasian Conference on Robotics and Automation*, Brisbane, Australia, 1-3 December 2010.

[J.Ko et al., 2010b] J.Ko and D.Fox, Learning GP-BayesFilters via Gaussian Process Latent Variable Models. *Springer Science + Business Media, LLC*, 2010.

[Rasmussen et al, 2005a] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. *The MIT Press*, 2005.

[C.Guestrin et al., 2005b] C.Guestrin, A.Krause and P.S.Ajit. Near-Optimal Sensor Placements in Gaussian Processes. *Proceedings of 22nd International Conference of Machine Learning*, Bonn, Germany, 2005.

[J.Ko et al., 2008] J.Ko and D.Fox. GP-BayesFilters using Gaussian Process Prediction and Observation Models. 2008 IEEE International Conference of Intelligent Robots and Systems, Nice, France, 2008.

[Plagemann et al., 2007] C.Plagemann, D.Fox and W.Burgard. Efficient Failure Detection on Mobile Robot using Gaussian Process Proposal. *Proceeding of The International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[A.Fod et al.,2002] A.Fod, A. Howard and M.J.Mataric. Laser-based People Tracking. 2002 IEEE International Conference on Robotics and Automation (ICRA), pages 3024—3029, Washington DC, May 2002.

[P.Chakravarty et al., 2005] P.Chakravarty and R.Jarvis. Multiple Target Tracking for Surveillance: A Particle Filter Approach. *2005 IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 181—186, Melbourne, Dec 2005.

[G.Shay, 2007] Geza Shay. Introduction to Probability with Statistical Applications. *Birkhauser Boston*, 2007.