# PERSONS TRACKING WITH GAUSSIAN PROCESS JOINT PARTICLE FILTERING

*Jaakko Suutala*[†*], *Kaori Fujinami*[‡], *and Juha Röning*[†]

[†]Dept. of Electrical and Information Engineering, P.O. Box 4500, 90014 University of Oulu, Finland
{jaska, jjr}@ee.oulu.fi
[‡]Tokyo University of Agriculture and Technology, Japan
fujinami@cc.tuat.ac.jp

**ABSTRACT**

This paper presents an approach to tracking persons using Gaussian Processes (GP) and Particle Filtering (PF). We used a binary switch sensor floor, which provides a natural and transparent way to build an indoor positioning and tracking system. However, it poses many challenges by producing nonlinear non-Gaussian measurements of true location. To solve these issues we present a novel algorithm. It uses PF for Bayesian tracking and data association combined with learned GP regression to correct estimates. Furthermore, the proposed algorithm, called Gaussian Process Joint Particle Filtering (GPJPF), handles multiple targets, where each particle models the targets' states jointly. To handle the data association problem and interaction between targets in close proximity, a Markov Random Fields (MRF) -based motion model was applied. Along with the GP model, it can be used directly as an additional factor when calculating the importance weights of particles. In comparison, the proposed method outperforms conventional Gaussian process and particle filtering methods.

## 1. INTRODUCTION

Tracking user positions from uncertain sensor information is very important when building smart, context-aware, and interactive environments [1] for example to be able to monitor daily routines or abnormal behavior, or to provide personalized services for users acting in an environment. A common approach employed in smart environments is to use video- and audio-based techniques [2, 3]. Although those sensors can provide rich information about users and the environment, vision-related systems could suffer from different lighting conditions and occlusion, and audio-related systems, from background noise, for instance. Another popular approach is to use WLAN positioning with mobile devices or RF, ultrasound, and infrared systems [1, 4]. The required wearable sensors can limit the user' s mobility or the user can easily forget to take the device along when acting in the environment.

This paper combines methods to presents novel algorithms and an alternative sensor approach to tracking persons in ubiquitous computing environments. Instead of placing individual sensors on every target in the environment, a floor sensor can be used as an environmental sensing system to recognize and locate active targets simultaneously. As a first step towards a smart environment, we concentrated on person localization and tracking using a sensory system installed on the floor surface.

*This work was supported by InfoTech Oulu Graduate School.

Quite similar to this work, Murakita et al. [5] presented a binary sensory system to track persons. It uses the Particle Filtering (PF) [6] technique to perform sequential position predictions using two different kinds of measurement models. However, instead of assuming known initial positions, we extended PF-based Bayesian filtering to a more general and practical approach that deals with multiple persons entering and leaving the sensor area at arbitrary time steps. We also present a flexible Gaussian Processes (GP) [7] -based model for learning to track human motion using displacement expert [8]. We derived an algorithm that combines state estimation with accurate GP-based correction and importance weighting steps. In addition, we extended our approach to multiple persons by adding a Markov Random Fields-based motion model [9] to handle interaction between persons as well as models for detecting persons entering and leaving the sensor area. Previously, GPs and PFs have been applied differently to tracking in computer vision [10, 11, 12, 13] and robotics [14, 15], for instance. To summarize, the contributions of the work are: combining Gaussian Process regression and Particle Filtering into a novel person tracking algorithm, extending the algorithm to handle a variable number of interacting persons entering and leaving the sensor area as well as applying the algorithm to a real-time tracking system using a novel floor sensor setting.

## 2. TRACKING METHODOLOGY

### 2.1. Gaussian Process Regression

Gaussian Processes (GP) [7] are a non-parametric probabilistic approach to learn kernel machines and have received a lot of attention in the machine learning community in recent years. They provide many useful advantages over Support Vector Machines (SVM) [16] and variants such as the ability to model uncertainty of estimates directly and perform model selection based on Bayesian inference.

The Gaussian process is a collection of random variables that have a joint Gaussian distribution. These random variables represent the value of the function $f(\mathbf{x})$ at a given location, and the GP is completely specified by its mean $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and covariance functions $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[f(\mathbf{x}) - m(\mathbf{x})] - m(\mathbf{x}')$, and they present the Gaussian process $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $\mathbf{x}$ and $\mathbf{x}'$ are two input feature vectors.

Let $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]^T$ be a training dataset of N x D dimensional input feature data matrix and $\mathbf{y} = [y_1 \dots y_N]^T$ an N x 1 dimensional vector of continuous targets. In the Gaussian process the regression output is modeled using a noisy version of function $\mathbf{y} = f(\mathbf{x}) + \epsilon$. Assuming the additive independent identically distributed Gaussian noise $\epsilon$, the posterior probability of latent functions is analytically solvable and leads to Gaussian predictive distribution $\overline{f}_* = \mathbf{k}_*^T(K + \sigma_n^2 I)^{-1}\mathbf{y}$ and $\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T(K + \sigma_n^2 I)^{-1}\mathbf{k}_*$, where $\overline{f}_*$ and $\mathbb{V}[f_*]$ are the mean and variance predictions of an unknown input example, respectively. $\mathbf{k}_*$ is the vector of covariances between the test example $\mathbf{x}_*$ and training examples, $K$ is the matrix of covariances between training examples $\mathbf{X}$, $k(\mathbf{x}_*, \mathbf{x}_*)$ is the covariance between a test example $\mathbf{x}_*$ and itself. $\mathbf{y}$ are the output targets in the training data set and $\sigma_n^2$ is noise variance.

The covariance function specifies prior knowledge and similarity between examples. Many different Mercel covariance functions producing positive semi-definitive kernel matrix are presented in the literature [7]. One of the most popular is the Squared Exponential (SE) (or Gaussian) covariance function $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2l^2}||\mathbf{x} - \mathbf{x}'||^2)$, where $\sigma_f^2$ is signal variance and $l$ is a length scale.

Typically, the covariance function will have some free parameters (such as $\sigma_f^2$, $l$). Training the GP regression model is to determine the values of the hyperparameters. Using the SE covariance function and independent noise variance $\sigma_n^2$ we can collect these hyperparameters into the common vector $\Theta = [\sigma_f^2, l, \sigma_n^2]$. The hyperparameters $\Theta$ can be learned by maximizing the log marginal likelihood (or evidence) $\log p(\mathbf{y}|X, \Theta) = -\frac{1}{2}\mathbf{y}K_y^{-1} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi$ of training data $D = (\mathbf{X}, \mathbf{y})$, where $K_y = K + \sigma_n^2 I$. This objective function can be optimized, for example, using gradient-based methods [7]. In position tracking we usually need to estimate more than one dimension, so a multi-output regression model needs to be implemented. We apply a coupled GP where the noise of each dimension is handled independently, but a block-diagonal covariance matrix with common hyperparameters is applied to model correlation between different dimensions by learning the hyperparameters from the data [17].

### 2.2. Learning the Displacement Expert

In online tracking applications we are interested in modeling dynamic events such as position transitions. GP regression can be trained to predict continuous outputs from input features. In visual tracking, Williams et al. [8] proposed an algorithm to train a displacement expert (i.e., the regressor) between a high-dimensional image space and a low-dimensional state space such as position, pose, and other continuous variables. As an alternative to predicting true position (or other variables), we can try to predict the difference between true position and some estimate. The advantage is that we do not need to collect a huge data set of training examples, but can use a small set of seed examples (e.g., images) and then sample the displacement of these examples, for example, from their uniform distribution $y \sim uniform(-\Delta, \Delta)$, where $\Delta$ is the displacement range from spatial location coordinates. Disadvantages are that compared to conventional Bayesian filtering, offline training period is needed. Moreover, in some applications it would be problematic to determine the true target position to collect training data.

Figure 1 presents an example pattern from the floor sensor studied in this work during single foot contact. In the case of two feet contact, we could set the true position to centre of mass point. To calculate features we can use, for example, a rectangle centered at the true position and sample the training examples from it. Let vector $\mathbf{u} = [u_v, u_h]$ present physical vertical $u_v$ and horizontal $u_h$ location coordinates on the floor. If we transform the measurements from the rectangle to input feature vector $\mathbf{x} = f(\mathbf{I}(\mathbf{u}))$ and the displacement to output target vector $\mathbf{y} = [\Delta u_v, \Delta u_h]$, we can learn the mapping between the input and output using the GP model presented in section 2.1. The algorithm presented in [8] can be used to collect a training data set by sampling from the seed examples and transforming the examples to feature vectors and the displacements to corresponding output values. Finally, the displacement expert, such as a GP regressor, can be learned from the data set.

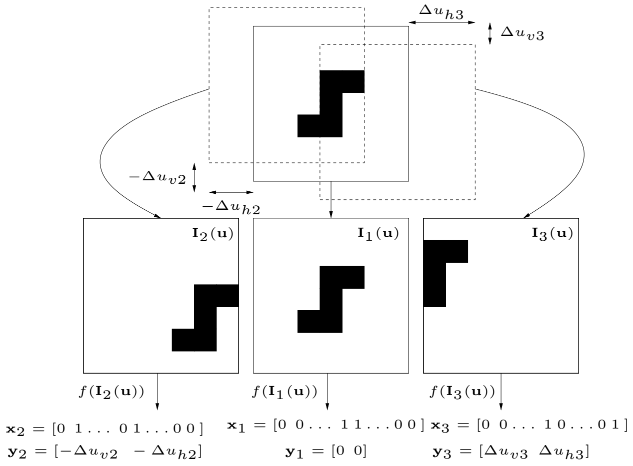



**Fig. 1**. Example procedure of sampling with displacements. The solid rectangle above is the region of interest of the black target. Two rectangles with dashed lines are sampled from the original region of interest, which leads to the spatial displacement regions below.

After the training, the GP model can be used to predict the displacement, and more interestingly in a tracking application, to estimate the current position $\mathbf{u}_t$ from the previous position $\mathbf{u}_{t-1}$ by $\mathbf{u}_t = \mathbf{u}_{t-1} + GP_\mu(\mathbf{x})$, where $\mathbf{x}$ is an input example. Now the prediction is based on the GP mean (i.e., point prediction) alone. The following section shows how the uncertainty estimation of GP (i.e., the variance of displacement) can be applied to sequential Bayesian filtering framework.

### 2.3. Joint Particle Filtering

Particle filtering (PF) [6] is an approximation method for nonlinear non-Gaussian dynamic sequential modeling, and it is very useful, e.g., in online tracking applications in environments with uncertain sensor measurements. There are many ways to extend particle filtering to multiple-target tracking. The simplest approach is to use multiple independent filters, one for each target. However, when the targets are close to each other and the measurements are noisy, independent filters lose their ability to keep the track of individual targets, and the target with the strongest measurements and best likelihood score will capture nearby targets. In [18] a mixture particle filter approach was developed. It uses an independent filter for each target. These components then form a mixture model where interaction between targets is handled by the mixture weights. The standard SIR particle filter can be embedded in the iterations, but a clustering method is needed to keep the mixture model updated. This could be problematic in settings where measurement of targets is multi-modal, sparse, and too similar over the group of targets.

Multiple target tracking can also be formulated using a Joint Particle Filter (JPF) [9] presentation where each particle captures the state $\mathbf{u}$ of all the targets jointly. Similarly to the mixture approach, we can sample from the motion model and set the likelihood score for each target independently $p(\mathbf{u}_t|\mathbf{u}_{t-1}) \propto \prod_i p(\mathbf{u}_t^i|\mathbf{u}_{t-1}^i)$. Moreover, likelihood scores can be calculated independently for each target and then used to form a factored likelihood model $p(\mathbf{z}_t|\mathbf{u}_t) \propto \prod_i p(\mathbf{z}_t^i|\mathbf{u}_t^i)$. The detailed presentation can be found in [9].

### 2.4. Markov Random Fields for Target Interaction

In multiple target tracking, the most problematic settings are related to cases where targets are currently located physically near to each other. This is known as a data association problem, where it is difficult to decide which target produces which of the measurements. Khan et al. [9] presented a multi-target interaction model applied to a Joint Particle Filter based on the Markov Random Fields (MRF) [19] motion model. Their application consisted of visual tracking multiple similar interacting targets, where the motion of individual targets is affected by the motion of nearby targets [9]. We applied a similar MRF model, but our goal was two-fold. First, the motions of interacting persons affect each other. Second, we can apply a more accurate measurement model to the data association problem (e.g., handling false alarm measurements).

MRF [19] is an undirected graph $G = (V, E)$, where random variables are presented as nodes (i.e., vertices $V$), and dependencies between nodes are presented as undirected edges ($E$). Joint probability is factored as a product of local potential functions at each node, and interactions are defined in neighborhood cliques. Following [9], we used pairwise MRF, where the cliques are pairs of nodes connected by the edge in the graph. The pairwise interaction potentials $\psi(\mathbf{u}^i, \mathbf{u}^j)$ are expressed by means of the Gibbs distribution in the log domain $\psi(\mathbf{u}^i, \mathbf{u}^j) \propto \exp(-g(\mathbf{u}^i, \mathbf{u}^j))$, where $g(\mathbf{u}^i, \mathbf{u}^j)$ is a penalty function and could be set using the degree of overlap when targets interact. When MRF is dynamically constructed at every time step $t$, the factored motion model becomes $p(\mathbf{u}_t|\mathbf{u}_{t-1}) \propto \prod_i p(\mathbf{u}_t^i|\mathbf{u}_{t-1}^i) \prod_{i,j \in E} \psi(\mathbf{u}_t^i, \mathbf{u}_t^j)$.

The MRF motion model can be directly embedded in the Joint Particle filter using the factored likelihood expression $w_t = w_{t-1} \prod_{i=1}^n p(\mathbf{z}_t^i|\mathbf{u}_t^i) \prod_{i,j \in E} \psi(\mathbf{u}_t^i, \mathbf{u}_t^j)$, where $w_t$ is the weight of the particle, $p(\mathbf{z}_t^i|\mathbf{u}_t^i)$ is the likelihood score of the $i$:th target, and $\psi(\mathbf{u}_t^i, \mathbf{u}_t^j)$ is the interaction term between targets $i$ and $j$, respectively.

## 3. GAUSSIAN PROCESS JOINT PARTICLE FILTERING

Using the tracking methodology presented in section 2, it is straightforward to combine these methods into a novel tracking algorithm, Gaussian Process Joint Particle Filtering (GPJPF). The proposed real-time tracking algorithm follows the standard phases of the Bayesian filter, and more specifically the Sampling Importance Resampling (SIR) Particle Filter [6]. As prior knowledge, we determined the motion model (i.e., how the states evolve over time) for where to sample at each time step to predict the target location.

Furthermore, we collected a training data set of feature vectors from regions of interest as well as the output targets of position displacements (cf. Figure 1). A discriminative probabilistic Gaussian Process regressor was trained between the measurements and the displacements from the true positions. The advantages of using machine learning are that we could use a simple motion model (e.g., prior linear Gaussian transition) and model possible non-linearities with the trained GP. It provides a prediction of displacement as well as an uncertainty measure (as a variance of displacement), which could be added directly to the correction step of the Bayesian filter, eliminating the need to build a measurement model separately.

Let $\mathbf{u}_t$ be the state estimate predicted using the motion model and $GP_\mu(\mathbf{x}) = \overline{f}_*$ and $GP_\Sigma(\mathbf{x}) = \mathbb{V}[f_*]$ be the mean and covariance of the predicted GP displacement of $\mathbf{x} = f(\mathbf{I}(\mathbf{u}_t))$ (or more precisely, of the region of interest centered at $\mathbf{u}_t$). The GP-based correction can be calculated as follows

$$\tilde{\mathbf{u}}_t = \mathbf{u}_t + GP_\mu(\mathbf{x}) \qquad (1)$$

and the update equation for the particle importance weighting $w_t$ becomes $w_t = w_{t-1}\mathcal{N}(\tilde{\mathbf{u}}_t; \mathbf{u}_t, GP_\Sigma(\mathbf{x}))$.

To handle multiple targets, and possibly a variable number of targets, we can extend the proposed tracking algorithm using a couple of more phases and the joint state presentation. Adding the factored likelihood presentation and MRF-based interaction, the importance weight update calculation can be presented as

$$w_t = w_{t-1} \prod_{i=1}^{n} \mathcal{N}(\tilde{\mathbf{u}}_t^i; \mathbf{u}_t^i, GP_\Sigma^i(\mathbf{x})) \prod_{i,j \in E} \psi(\tilde{\mathbf{u}}_t^i, \tilde{\mathbf{u}}_t^j) \qquad (2)$$

where $GP_\mu^i(\mathbf{x}) = \tilde{\mathbf{u}}_t^i - \mathbf{u}_t^i$, $GP_\mu^i(\mathbf{x})$, and $GP_\Sigma^i(\mathbf{x})$ are the displacement GP mean and covariance of the $i$:th target in particle $\mathbf{u}_t$ and $\psi(\tilde{\mathbf{u}}_t^i, \tilde{\mathbf{u}}_t^j)$ is the MRF interaction term between GP-corrected targets $i$, and $j$, respectively. GP corrections are calculated independently for each target, similar to Eq. 1.

## 4. GPJPF TRACKING ON SENSOR FLOOR

A VS-SF55 InfoFloor sensor system made by Vstone Corporation (in Japan) was used in this work. The system contains 12 50 cm x 50 cm sensor tiles. Each tile includes 25 10 cm x 10 cm binary switch sensors. A 3 $m^2$ area was covered by altogether 300 sensors. The sensors use diode technology and are able to detect a minimum weight of 200-250 $g/cm^2$ on the surface. Data were collected from each sensor using a 16 Hz sampling rate.

In a multiple-person tracking scenario, the sensor setting faces many challenges. When the persons are producing multiple patterns with both feet (see Figure 5) and are physically near to each other in the sensor area, it is difficult to determine the center point of each person. We need to associate each measurement pattern with a different person. To overcome this problem, we can use the proposed method where part of the tracking technique is learned from the data, and prior knowledge of the persons' previous positions are applied to a Bayesian temporal filtering framework.

The state space model in our system is a simple stationary first-order Markov process, where target state $p(\mathbf{u}_t|\mathbf{u}_{t-1}) \sim \mathbf{u}_t$ is approximated from the targets previous state $\mathbf{u}_{t-1}$. The extension to constant velocity or more advanced motion model is straightforward. Following the expression of the Gaussian measurement model, we can apply GP prediction to importance sampling. Let $GP_\mu = GP_\mu(\mathbf{x})$ represent the GP mean displacement of $\mathbf{x}$, centered at particle $\mathbf{u}$, and the $d$ x $d$ dimensional GP displacement covariance matrix is $GP_\Sigma = GP_\Sigma(\mathbf{x})$. The GP-based measurement model becomes

$$p(\mathbf{z}_t|\mathbf{u}_t) = \frac{1}{(2\pi)^{d/2}\sqrt{|GP_\Sigma|}} \exp[-\frac{1}{2}(GP_\mu^T GP_\Sigma^{-1} GP_\mu)]. \quad (3)$$

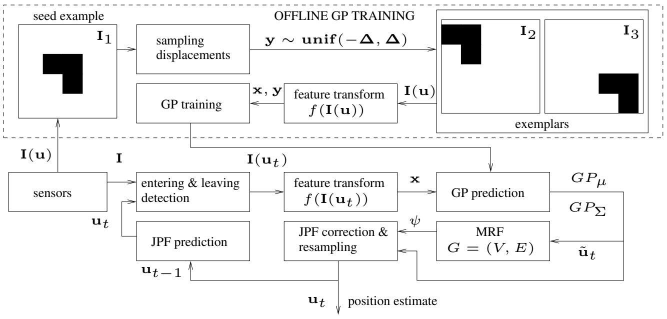



**Fig. 2**. A flowchart of the GPJPF tracking system.

Novel targets are recognized using current measurements (clustering center-of-mass points of connected components on the floor) corrected with the GP model and existing particles by calculating distances between the spatial center of particle set clusters and position candidates. If the candidate point distance is not less than the given threshold from existing particles, a novel target position is initialized. When an existing target leaves the sensor area, the particle component is removed from the joint presentation. If the prediction of the current particles is far from the measurements (i.e., larger than a given threshold), it is deleted from each particle. When two (or more) targets interact (or walk near to each other), the current particle distribution can overlap and discrimination between targets is impossible. In such a case we can use the interaction potentials of the MRF to re-weight the particles by calculating the potentials between nearby targets. Particles in the non-overlapping area are given more weight and more probably survive after the resampling step, whereas interacting particles in overlapped area are discarded. Flowchart of the GPJPF tracker is presented in Figure 2.

## 5. RESULTS

### 5.1. Experimental Settings

In this study we compare the proposed method to three other sampling based methods and their GP-based variants. The first method was Independent Particle Filtering (IPF), where each target is modeled with a single independent particle set as well as its GP variant (GPIPF). The second method was Mixture Particle Filtering (MPF) [18] where each target is modeled with an independent particle set component, but the targets interact via a common mixture weight presentation. We modified the original algorithm by removing the splitting and merging steps (see [18]) and adding the MRF-based motion model. Furthermore, its GP variant (GPMPF) was examined. Finally, Joint Particle Filtering (JPF) and the proposed Gaussian Process Joint Particle Filtering (GPJPF) were tested. In these methods each particle represented the state space of every target being tracked.

A single GP model was trained from 4 persons' data sets of 50 examples. The examples were sampled from the walking sequences performed by the each subject at once, and no training data from actual multi-person walking sequences were used. A 60 cm x 60 cm region of interest was used, providing a 6 x 6 feature area and a 36-dimensional input feature vector when using 10 cm x 10 cm sensor tiles. A squared exponential kernel was applied and the hyperparameters were trained using the marginal likelihood. The stationary motion model was applied by setting the noise variances to 20 cm. The Gaussian measurement model with 60 cm noise was applied to conventional PFs, and GP-based PFs were equipped with a GP-driven measurement/correction steps in Equations 1 and 3. In the entering and leaving models the appearance and disappearance probability thresholds were set at 0.0 and 1.0. For the MRF interaction terms, similar to [9], the linear interaction function $\gamma p$ was used. $p$ is the area of overlapping between two targets and was set at 2.0. In later experiments we studied the influence of interaction by changing the value of $\gamma$.

To test and compare the different methods, altogether 70 walking sequences, including 8539 data frames, were collected from 2 male and 1 female subjects. In each sequence two different walkers from the group of 3 subjects walked. There were altogether 7 different walking settings, which were repeated 10 times each. The walking paths included different individual directional changes, different starting and ending positions, and arbitrary entering and leaving times. In addition, different interactions - meetings, followings, and passing by situations - were experimented with. In these data sequences the minimum distances between targets were varied from 30 cm to 150 cm. Moreover, to test the proposed methods, one longer data sequence (1255 frames), which included non-predefined walking paths and natural interaction, was performed by 1 male and 1 female subject simultaneously.

### 5.2. Comparison of Algorithms

Table 1 presents a comparison of these 6 different particle filtering methods using the 70 test sequences described above. Each sequence was repeated 3 times to avoid random effects on initialization and sampling of filters. The results are presented using two different failure rates. First is sequence failure, which measures if the tracker failed to keep the true identity, position, and the number of targets through the whole sequence. These were observed manually from the visualization of the tracking simulation. Second is frame failures, which measures different failures in each frame. These include position failure, which was set at 60 cm, similar to the previous section. Identity and number of failures measure if there are wrong identities (i.e., different than the two persons who entered in the sensor area) and a wrong number of targets (i.e., different than one or two persons in these tests) detected. Additionally, total frame failures, which measures if at least one of the three failure types (i.e., position, identity, or number of targets) has occured at the particular time step, are given. The results indicate that simple independent filters are not able to keep track of multiple persons, but the target with the strongest measurements and the best likelihood score will capture the nearby target. A GP-based particle filter outperforms conventional particle filters, showing better discriminative power, and GPJPF outperforms all other methods, showing the best performance when joint state presentation, MRF motion model, and GP model are combined.

### 5.3. Discrimination Accuracy

Next we tested the discrimination accuracy of the different methods. We took 60 of the 70 sequences described above in which interaction happened and calculated the minimum distance between the targets in each sequence. We divided the sequences into different distance gaps and calculated histograms of discrimination failures. The failure rates are calculated from these histograms. Similar to the previous tests, these tests were completed 3 times

| Method | Samples | Sequence failures (%) Total | Frame failures (%) Position | Identity | Number | Total |
|---|---|---|---|---|---|---|
| IPF | 50/target | 57.14 | 15.18 | 6.00 | 7.18 | 16.58 |
| GPIPF | 25/target | 52.86 | 13.65 | 5.10 | 6.92 | 15.28 |
| MPF | 50/target | 12.38 | 0.37 | 0.37 | 0.87 | 1.11 |
| GPMPF | 25/target | 8.57 | 0.21 | 0.25 | 0.28 | 0.48 |
| JPF | 100 | 9.05 | **0.04** | **0.00** | 0.47 | 0.51 |
| GPJPF | 50 | **3.81** | 0.09 | **0.00** | **0.06** | **0.12** |

**Table 1**. Tracking results of two persons using different methods. The smallest failure rates in each category are highlighted.

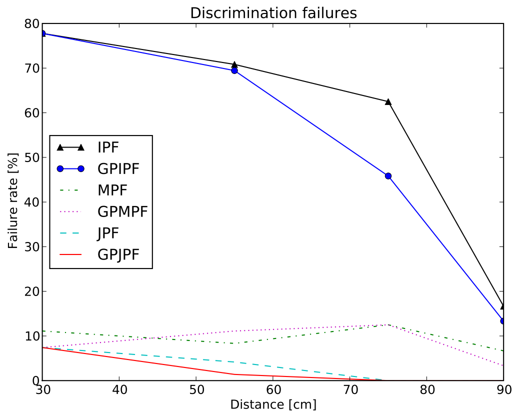



**Fig. 3**. Failure rates of keeping track of two persons when the distance between them changes.

in each sequence. Same model parameters from the previous experiments were used. Figure 3 shows the failure rates when the distance between persons is increased. GPJPF and IPF are most accurate, showing failure rates below 8% when the distance is 30 cm - 55 cm. When the distance is more than 75 cm, both are able track persons perfectly, showing 100% accuracy. GPJPF is slightly better overall. The mixture filters perform with around 10% failure rates and the independent filters are not accurate until the distance is more than 90 cm, even then showing failure rates of more than 10%.

### 5.4. Influences of the MRF Model

Finally, the effect of the MRF motion model was tested by changing the interaction level. We compared the two best methods, GPJPF and JPF, by changing the $\gamma$ parameter. We used a long data sequence of two simultaneously walking persons, including a lot of interaction and small distances between persons. Each method was repeated 5 times for each interaction level. Figure 4 shows the frame-based failure rates of the different $\gamma$ parameters. When $\gamma = 0.0$, the MRF is ignored. First the results show that the MRF model is very important, and when it is totally ignored the total failure rates are 79.6% and 55.6% for JPF and GPJPF, respectively. Second, JPF is more sensitive to the lack of interaction, showing that the GP-driven measurement model has more discriminative power when the targets are physically close to each other. Figure 5 shows some frames from the test data sequence when two persons were tracked using GPJPF.

## 6. CONCLUSIONS

This paper proposed novel combination of algorithms for tracking persons. In the example application, binary switch floor sensors were used to detect walking persons. The proposed tracking algorithm is based on Gaussian Process (GP) regression learned from the training data to predict the spatial displacement of the tracked person, as well as on Particle Filtering (PF), which is used to smooth the estimates and handle multi-modal distributions produced by the different types of foot contacts on the floor. Compared with a conventional particle filter, no hand-tuned measurement model (and noise variances) are needed; they are automatically learned from the data using optimization of the marginal likelihood in terms of noise variance and covariance hyperparameters. These are important properties, because it is difficult to build a measurement model that is able to model different variations in sparse multi-modal measurements like in our floor sensor-based application. In addition, the algorithm was extended to track multiple simultaneous walkers, handling a person entering and leaving the sensor area, and to model interaction between persons, which are both practically important when building real-life applications. The presentation is based on the Joint Particle Filter approach, where each state represents the positions of all the current walkers. The tracking and entering/leaving are handled using GP and PF. The interaction, and more precisely the data association, problem between adjacent targets is handled using a Markov Random Fields (MRF) motion model by giving less weight to uncertain particles on the overlapping area between persons. The GP and MRF models as well as joint presentation can be applied directly to the standard SIR particle filtering framework showing superior tracking accuracy compared to conventional methods.

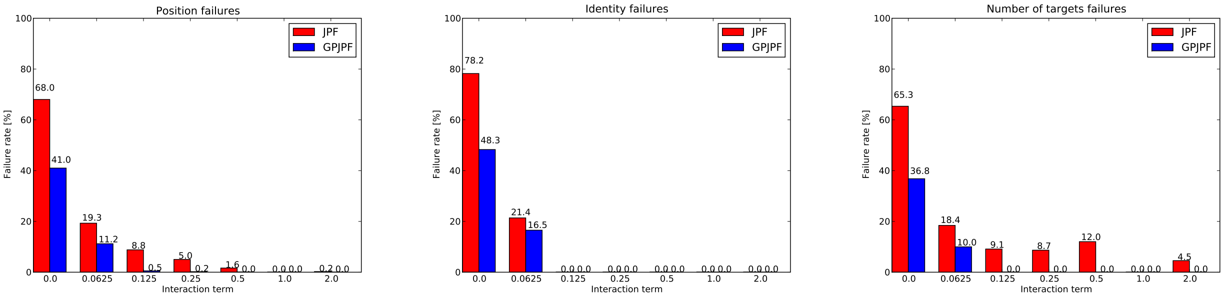
## 7. REFERENCES


[1] M. Hazas, J. Scott, and J. Krumm, "Location-aware computing comes of age," *IEEE Computer Magazine*, vol. 2, pp. 95–97, February 2004.

[2] J. Krumm, S. Harris, B. Meyers, B. Brummit, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Third IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 1 2000.

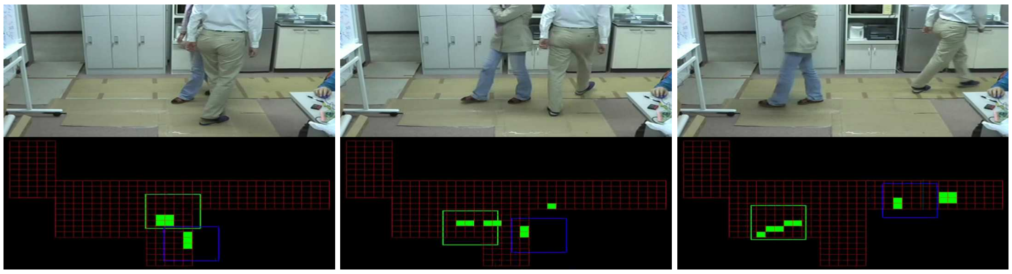
[3] N. Checka, K.W. Wilson, M.R. Siracusa, and T. Darrell, "Multiple person and speaker activity tracking with a particle filter," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, May 17-21 2004, pp. 881–884.

[4] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitous computing: A case study," in *Proceedings of the Sixth International Conference on Ubiqui-*




(a) Position failures. (b) Identity failures. (c) Number of targets failures.

**Fig. 4**. Tracking failure rates of different Markov Random Fields interaction levels.



(a) Frame 434. (b) Frame 445. (c) Frame 462.

**Fig. 5**. Data frames captured from a 1522-frame-long sequence when Gaussian Process Joint Particle Filtering is used to track two persons.


*tous Computing (Ubicomp)*. September 2004, pp. 88–106, Springer-Verlag.

[5] T. Murakita, T. Ikeda, and H. Ishiguro, "Human tracking using floor sensors based on the markov chain monte carlo method," in *Proceedings of Seventeenth International Conference on Pattern Recognition (ICPR)*, Cambridge, UK, August 2004, pp. 917–920.

[6] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, 2001.

[7] C. Rasmussen and C. Williams, *Machine Learning for Gaussian Processes*, The MIT Press, USA, 2006.

[8] O. Williams, A. Blake, and R. Cipolla, "Sparse bayesian learning for efficient visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no. 8, pp. 1292–1304, 2005.

[9] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, pp. 1805–1918, 2005.

[10] J.M. Wang, D.J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis Machine Intelligence (TPAMI)*, vol. 30, no. 2, pp. 283–297, 2008.

[11] J. Wang, Y. Yin, and H. Man, "Multiple human tracking using particle filter with gaussian process dynamical model," *EURASIP Journal on Advances in Signal Processing*, vol. volume 2008, pp. 10 pages, 2008.

[12] L. Raskin, E. Rivlin, and M. Rudzsky, "Using gaussian process annealing particle filter for 3d human tracking," *EURASIP Journal on Advances in Signal Processing*, vol. volume 2008, pp. 13 pages, 2008.

[13] H. Li, Y. Wu, and H. Lu, "Visual tracking using particle filters with gaussian process regression," in *3rd Pacific Rim Symposium on Advances in Image and Video Technology (PSIVT)*. 2009, pp. 261–270, Springer-Verlag.

[14] C. Plagemann, D. Fox, and W. Burgard, "Efficient failure detection on mobile robots using particle filters with gaussian process proposals," in *The Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

[15] J. Ko and D. Fox, "GP-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots Journal*, 2009.

[16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, UK, 2000.

[17] O. Williams, *Bayesian Learning for Efficient Visual Inference*, Ph.D. thesis, University of Cambridge, UK, 2005.

[18] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multimodality through mixture tracking," in *Ninth IEEE International Conference on Computer Vision (ICCV)*, 2003.

[19] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, Springer-Verlag, 2009.