

# QUANTITATIVE PERFORMANCE EVALUATION OF A PEOPLE TRACKING SYSTEM ON A MOBILE ROBOT

*Grzegorz Cielniak<sup>1</sup>, André Treptow<sup>2</sup> and Tom Duckett<sup>1</sup>*

<sup>1</sup>AASS, Department of Technology, University of Örebro, Örebro, Sweden

<sup>2</sup>Department of Computer Science, University of Tübingen, Tübingen, Germany

## ABSTRACT

Future service robots will need to keep track of the persons in their environment. A number of people tracking systems have been developed for mobile robots, but it is currently impossible to make objective comparisons of their performance. This paper presents a comprehensive, quantitative evaluation of a state-of-the-art people tracking system for a mobile robot in an office environment, for both single and multiple persons.

## 1. INTRODUCTION

As research robots move closer towards real applications in populated environments, these systems will require robust algorithms for tracking people to ensure safe human-robot cohabitation. The tracking problem is well-known in computer vision, where there are developed systems used mostly in surveillance and identity verification. In the case of mobile robots, these tasks become even more challenging since the robot is moving and the environment is unpredictable. Systems that can be used on mobile robots should be robust (increased noise), adaptive (unknown environment), fast (real-time) and non-invasive (normal human activity is unaffected). Thus many methods from computer vision cannot be applied directly to mobile robots. Existing people recognition systems on mobile robots use information from range sensors [1], cameras [2, 3] or a combination of different sensors [4]. However, at present it is impossible to compare the performance of these methods, due to the lack of a commonly accepted methodology for quantitative performance evaluation.

The problem of evaluating tracking systems has been addressed recently by the computer vision community [5]. The consensus is that there is no single metric that could indicate sufficiently the quality of the entire system. For a proper evaluation it is important to use different metrics quantifying different performance aspects of the system. Examples of different metrics can be found in [6, 7, 8, 9]. Having a good set of performance measures allows to optimise algorithm parameters, check performance of the tracker for

different kinds of data, quantitatively compare different algorithms, support development of the algorithm, and decide upon trade-offs between different performance aspects. The evaluation procedure requires that ground truth information is available. In the case of video data this is often a difficult, monotonous and labor demanding process. There were attempts to improve and automate this process by using some other algorithm to roughly select regions of interest that are refined later by hand [10], synthesised ground truth data [8], or systems performing fully automatic evaluation based on color and motion metrics [9]. One example of a system providing a set of tools for labelling ground truth data and metrics for evaluation of results is The Video Performance Evaluation Resource (ViPER) [6].

This paper presents a thorough evaluation of a people tracking system for mobile robots using ViPER.

## 2. EXPERIMENTAL SET-UP



Figure 1: PeopleBot equipped with a thermal camera in a populated corridor.

Our experimental platform was an ActivMedia PeopleBot mobile robot (Fig. 1) equipped with an array of sen-

sors including a thermal camera (Thermal Tracer TS7302, NEC). The camera can detect infrared radiation and convert this information into an image where each pixel corresponds to a temperature value (see Fig. 2). In our experiments the visible range on the gray-scale image was equivalent to the temperature range from 24 to 36 °C.

The robot was operated in an unconstrained indoor environment (a corridor and lab room at our institute). Persons taking part in the experiments were asked to walk in front of the robot while it performed two different autonomous patrolling behaviours: corridor following (based on sonar readings) and person following (using information from the implemented tracker), or while the robot was stationary. At the same time, image data was collected with a frequency of 15Hz. The resolution of the thermal images was  $320 \times 240$  pixels.

### 3. PEOPLE TRACKING SYSTEM

Most vision-based people recognition systems concern non-mobile applications e.g., surveillance or identity verification systems, where detection of persons can be solved easily by background subtraction methods that cannot be applied to mobile systems. Existing vision-based people recognition systems on mobile robots typically use skin colour and face detection algorithms. However these approaches assume the frontal pose of a person and require that the person is not too far from the robot. Thermal vision helps to overcome some of the problems related to colour vision sensors since humans have a distinctive thermal profile compared to non-living objects.

To track a person in the thermal image we use a Particle Filter and a simple elliptic model which is very fast to calculate. Particle Filters [11], such as the well-known Condensation algorithm [12] in the field of computer vision, provide a solution to the state estimation problem. The popularity of these methods is due to the fact that they can be used in the case of system non-linearities and multi-modal distributions.

The probability  $p(X_t|Z_t)$  of a system being in the state  $X_t$  given a history of measurements  $Z_t = \{z_0, \dots, z_t\}$  is approximated by a set of  $N$  weighted samples:

$$S_t = \{x_t^{(i)}, \pi_t^{(i)}\}, i = 1 \dots N. \quad (1)$$

Each  $x_t^{(i)}$  describes a possible state weighted by  $\pi_t^{(i)}$  which is proportional to the likelihood that the system is in this state. Particle Filtering consists of three main steps:

1. Create new sample set  $S_{t+1}$  by resampling from the old sample set  $S_t$  based on the sample weights  $\pi_t^{(i)}, i = 1 \dots N$
2. Predict sample states based on the dynamic model  $p(x_{t+1}^{(i)}|x_t^{(i)}), i = 1 \dots N$

3. Calculate new weights by application of the measurement model:  $\pi_{t+1}^{(i)} \propto p(z_{t+1}|X_{t+1} = x_{t+1}^{(i)}), i = 1 \dots N$ .

The estimate of the system state at time  $t$  is the weighted mean over all sample states:

$$\hat{X}_t = E(S_t) = \sum_{i=1}^N \pi_t^{(i)} x_t^{(i)}. \quad (2)$$

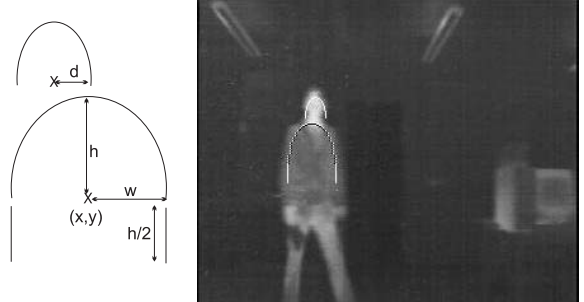


Figure 2: The elliptic measurement model used to track people in thermal images.

For each sample we use an elliptic contour measurement model to estimate the position of a person in the image: one ellipse describes the position of the body part and one ellipse measures the position of the head part. Therefore, we end up with a 9-dimensional state vector:  $x_t = (x, y, w, h, d, v_x, v_y, v_w, v_h)$  where  $(x, y)$  is the mid-point of the body ellipse with a certain width  $w$  and height  $h$ . The height of the head is calculated by dividing  $h$  by a constant factor. The displacement of the middle of the head part from the middle of the body ellipse is described by  $d$ . We also model velocities of the body part as  $(v_x, v_y, v_w, v_h)$ . The elliptic contour model can be seen in figure 2.

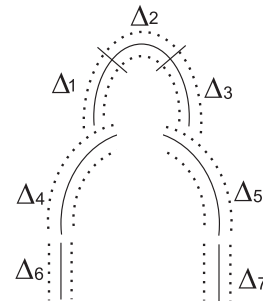


Figure 3: Elliptic model divided into 7 sections.

To calculate the weight  $\pi_t^{(i)}$  of a sample  $i$  with state  $x_t^{(i)}$  we divide the ellipses into different regions (see figure 3)

and for each region  $j$  the image gradient  $\Delta_j$  between pixels in the inner part and pixels in the outer part of the ellipse is calculated. The gradient is maximal if the ellipses fit to the contour of a person in the image data. The weight  $\pi_t^{(i)}$  is then calculated as the sum of all gradients multiplied by a weight factor:

$$\pi_t^{(i)} = W \cdot \sum_{j=1}^m \Delta_j \quad (3)$$

with

$$W = \sum_{j=1}^m w_j, \quad w_j = \begin{cases} 0 & : \text{if } \Delta_j < t_h \\ \omega_j & : \text{otherwise} \end{cases} \quad (4)$$

The value  $t_h$  defines a gradient threshold and the weights  $\omega_j$  are chosen in a way that the shoulder parts have lower weight to minimize the measurement error that occurs due to different arm positions.

The dynamic model that we use for the Particle Filter is a simple random walk: we model a movement with constant velocity plus small random changes. This value depends heavily on the speed of changes in the sensor data which is affected both by the frequency of the sensor and person’s dynamic. Our approach to track the contour of a person in the image is similar to the work by Isard and Blake [12] for tracking people in a grey image. However, they use a spline model of the head and shoulder contour which cannot be applied in our case because in situations where the person is far away or visible in a side view, there is no recognisable head-shoulder contour. The elliptic contour model is able to cope with these situations. The second advantage of using our contour model is that it can be calculated very quickly due to the fact that we measure only differences between pixel values on the inner and outer part of the ellipse.

The above considerations concern a single person scenario. When multiple persons appear on the scene another issue arises since it is not clear which observation corresponds to which person (the so-called data association problem), and persons may also occlude each other. In our case we used a set of independent particle filters to track different persons. In our system, a detection set of  $N$  randomly initialized particles is used to “catch” a new person entering the scene. Detection occurs when the average fitness of the particles exceeds a certain threshold in a few (3) consecutive frames. An independent particle filter is then assigned to each detected person, so the total number of particles used is  $(1 + M) \cdot N$ , where  $M$  is the number of persons detected. To avoid multiple detections in the same or similar regions the weight of detection particles is penalised in the case where they cross already detected areas. Methods based on independent trackers are computationally inexpensive but suffer in cases when tracked objects are too close. To reduce these problems we penalise the weight of those particles that intersect with other regions.

## 4. EVALUATION

This section presents the evaluation methodology used in our experiments. First we describe the format and process of acquiring ground truth data, then the metrics used for evaluation and finally experimental results.

### 4.1. Ground Truth

Our system was tested on the data collected by the robot during several runs. The acquisition procedure was planned such that it covered a diverse variety of possible scenarios including different persons with different behaviours and also different behaviours of the robot (45 tracks using person following, 13 tracks using corridor following and 7 tracks with a stationary robot). In total we obtained 65 different tracks including 17 different persons (11269 images with at least one person and 14059 images in total). To obtain the ground truth data we used results from a flood-fill segmentation algorithm corrected afterwards by hand using the ViPER-GT tool [6]. In our case we considered only a bounding box around a person. The top and bottom edges were determined from the contours of the head and feet while the sides were specified by the maximum width of the torso (without arms). The cases when persons appeared too close ( $< 3m$ ) or too far ( $> 10m$ ) to the robot were not taken into account. This type of ground truth information is just an approximation, and the quality of this process is affected by factors such as the natural blurred appearance of a person in the image, noise caused by the movement of the robot and also the skill of the person labelling the data. The size of the bounding box generated by the system was specified as  $2 * width$  and  $3 * height$  of the elliptic model, which is an approximation to the proportions of the human body.

### 4.2. Performance Metrics

There are several different metrics that can be used to evaluate a tracking system and the best way is to combine them. Here we present metrics that indicate temporal correspondence of the tracks and quality of detection and localisation. Detection and localisation metrics are specified for a single frame and the final results are weighted average values calculated for all frames. We refer to ground truth objects as target objects and objects detected by the algorithm as candidate objects. All these metrics were calculated by the ViPER-PE tool set [6].

#### 4.2.1. Temporal Metrics

In this case we consider time ranges of candidate and target objects. Time range is defined as a consistent sequence of

detections forming one track. For a given pair of the target range  $R_T$  and candidate range  $R_C$  we can calculate the following metrics:

- *Overlap coefficient (OC):*

$$OC = \frac{|R_T \cap R_C|}{|R_T|}. \quad (5)$$

This metric measures the fraction of frames in the target range which are also in the candidate range.

- *Dice coefficient (DC):*

$$DC = \frac{2 * |R_T \cap R_C|}{|R_T| + |R_C|}. \quad (6)$$

This metric indicates the number of frames that the target and candidate ranges have in common but also penalises any extra frames that are not common.

#### 4.2.2. Detection Metrics

These metrics take into account all candidate object detections regardless of how well they match a target. For each frame the number of target objects is denoted by  $N_T$  and number of candidate objects by  $N_C$ .

- *Object Count Recall (OCR):*

$$OCR = \frac{N_C}{N_T}. \quad (7)$$

This metric indicates how well an algorithm counts objects.

- *Object Count Precision (OCP):*

$$OCP = \frac{N_T}{N_C}, \quad (8)$$

is a counter-metric to *OCR* and corresponds to the false detection ratio.

- *Object Count Accuracy (OCA):*

$$OCA = \frac{2 * \min(N_T, N_C)}{N_T + N_C} \quad (9)$$

This metric is applied to check the accuracy of detection. It penalises both missing recalls (false negatives) and false alarms (false positives).

#### 4.2.3. Localisation Metrics

These metrics express relations between areas corresponding to candidate objects  $A_C$  and target objects  $A_T$ . The final result is a weighted average of all frames.

- *Object Area Recall (OAR):*

$$OAR = \frac{|A_T \cap A_C|}{|A_T|}. \quad (10)$$

This metric measures the proportion of each target object area covered by the corresponding candidate object. Recall is calculated for each target object and averaged for the whole frame.

- *Box Area Precision (BAP):*

$$BAP = \frac{|A_T \cap A_C|}{|A_C|}. \quad (11)$$

This metric is a counter-metric to *OAR* and it examines areas of candidate objects instead. Precision is computed for each candidate object and averaged for the whole frame.

- *Area Dice Coefficient (ADC):*

$$ADC = \frac{2 * |A_T \cap A_C|}{|A_T| + |A_C|}. \quad (12)$$

This metric is equivalent to the temporal dice coefficient (*DC*). This metric measures how well an algorithm covers the target object areas but also penalises areas that are not common.

All of the mentioned metrics are normalised to give percentages. If the nominator is greater than denominator the result is 100. If the denominator is 0 then the result is undefined.

### 4.3. Results

We checked the performance of the system with respect to different parameters including number of particles, percentage of samples used in the resampling step and the percentage of best samples used for calculating the weighted mean. As default values we chose 1000 particles, 90% of re-initialised samples and 30% of best samples used for calculation of the weighted mean. Tables 1, 2 and 3 show percentage results using the default values with one of the parameters chosen as an independent variable.

Table 1 shows the results for different numbers of samples (particles) for the single person case. The quality of tracking increases with the number of samples and satisfactory results can be obtained with 300 particles. With more than 1000 samples the quality of tracking saturates and there is no significant improvement in results. With less than 200 samples the tracker often loses tracks and estimates about position are inaccurate.

To reduce degeneracy of particles a proportion of all samples are re-initialised at every step of the particle filter.

Metric	Number of particles						
	100	200	300	500	1000	2000	5000
OC	83.8	90.3	92.5	93.1	93.2	93.5	93.8
DC	91.0	94.8	96.0	96.4	96.4	96.6	96.8
OCR	86.0	93.3	95.2	95.6	95.9	96.1	96.4
OCP	98.7	98.3	98.4	98.5	98.3	98.3	98.4
OCA	92.2	96.4	97.3	97.5	97.6	97.7	97.8
OAR	58.0	64.7	65.9	66.7	67.4	68.2	68.4
BAP	81.5	81.0	80.9	80.6	80.4	80.3	80.4
ADC	65.8	70.2	71.1	71.7	72.2	72.7	72.9

Table 1: Performance results for different number of particles (a single person case).

Metric	Percent of all particles						
	95	90	85	80	70	60	50
OC	93.2	93.2	93.1	93.1	92.4	90.1	84.2
DC	96.4	96.4	96.4	96.4	96.0	94.7	90.6
OCR	96.0	95.9	95.8	95.6	94.7	92.4	85.8
OCP	98.3	98.3	98.4	98.4	98.6	98.7	99.0
OCA	97.6	97.6	97.6	97.5	97.1	95.9	91.6
OAR	67.2	67.4	67.9	67.3	66.7	65.4	61.5
BAP	80.2	80.4	80.4	81.1	81.4	81.7	82.0
ADC	71.9	72.2	72.6	72.4	72.2	71.5	68.9

Table 2: Performance results for different percentage of resampled particles (a single person case).

The rest of the particles are used in the resampling step, and this proportion was a second parameter in our experiments. Results in Table 2 indicate low sensitivity of the tracker to changes of this variable. Values below 70% are not advisable and the best results were obtained for 85% of all samples. This means that the effects of degeneracy of particles play a minor role in our system. To increase robustness of the system to outliers, instead of calculating estimates from all samples we chose a proportion of the samples with the best weights. This variable was a third parameter and respective results are presented in Table 3. A smaller number of samples used for calculating the weighted mean increases tracker robustness but the quality falls in cases of less than 20%.

We also checked the system for tracking multiple persons. Figure 5 gives an overview of the data set used in our experiments. It consists of a sequence with multiple persons (up to four at the same time) that enter and leave the scene, interact and occlude each-other. We used the same default parameters as in the single person case with the exception that 1000 particles were used as a detection set and 1000 additional particles were used to track each person detected.

In comparison with the single person case there is a sig-

Metric	Percent of all particles						
	5	10	15	20	30	40	50
OC	93.4	93.7	93.8	93.5	93.2	92.1	88.3
DC	96.5	96.7	96.8	96.6	96.4	95.8	93.5
OCR	98.3	98.3	98.1	97.9	97.6	97.0	94.7
OCP	98.0	97.7	97.1	96.6	95.9	94.6	90.5
OCA	96.9	97.4	97.8	98.0	98.3	98.5	98.7
OAR	66.6	67.8	67.7	67.7	67.4	66.5	64.2
BAP	77.4	78.2	78.9	79.7	80.4	81.3	81.8
ADC	70.0	71.3	71.5	71.9	72.2	72.0	70.9

Table 3: Performance results for different percentage of particles with the best weights used for calculating state estimates (a single person case).

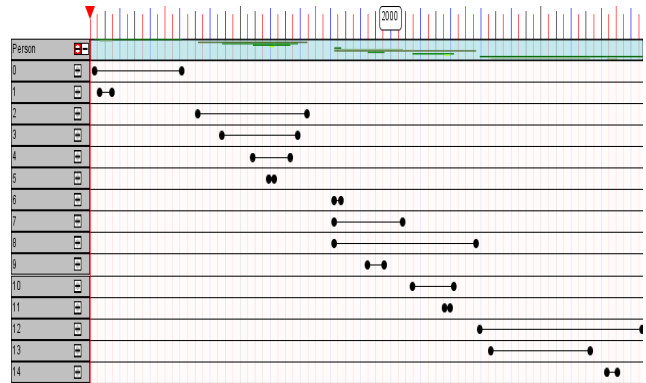


Figure 4: Time ranges of different tracks of persons in a multi-person sequence. The vertical axis corresponds to the number of a person and horizontal axis to a time stamp in the sequence (time range between 1-3677). This figure was extracted from the VIPER-GT tool.

nificant deterioration in the performance of the system (see results in Figure 5) due to fragmentation of the tracks caused by crossings and occlusions and the increased number of persons. Despite this the tracker detection performance is satisfactory. Slightly decreased detection rates are due to the sequential nature of the detector.

Generally the tracking system based on the thermal appearance of a person minimizes false alarms especially well. The performance of localisation is affected strongly by the fact that we are considering bounding boxes around a person, which results in low recall values especially in the case of distant persons. The use of a fast-to-calculate elliptic model results in low computational requirements. One iteration of the tracking algorithm using 300 particles on an AthlonXP 1600 processor requires only 11ms.

Metric	Result
OC	79.0
DC	85.3
OCR	90.9
OCP	95.2
OCA	92.8
OAR	51.7
BAP	83.5
ADC	56.8

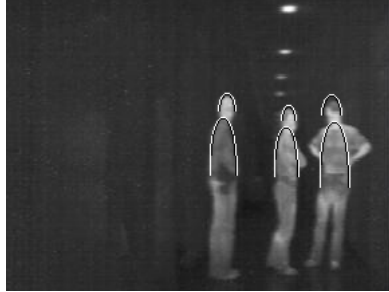


Figure 5: Left figure: performance results obtained in the multi-person case. Right figure: a thermal image with three detected persons.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a vision-based people tracking system for mobile robots and provided a thorough evaluation of its performance. The system uses a robust and fast tracking method based on a particle filter and a thermal contour model. We determined the optimal values of different system parameters. The quality of tracking depends on the number of particles used. The presented results allow for further investigation of the proposed system but also for comparison with other systems or data sets. The results indicate a good detection performance and consistent tracking in the case of single persons. The accuracy of localisation could be improved by using a more sophisticated model incorporating other features (e.g., based on the integral image). The tracker is also able to detect and track multiple persons. The performance of the tracking system here depends heavily on the intensity of interaction between persons (crossings and occlusions). The tracker tends to easily lose the track in such cases but it recovers quickly from tracking failures. Other multi-target tracking methods could improve the tracking quality, but due to the limitations of the thermal model (similar appearance of different persons in the thermal image), the problem of consistency of tracks under occlusion would remain. One way to solve this problem would be to incorporate other cues such as colour appearance models of persons.

## 6. REFERENCES

- [1] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2001.
- [2] Z. Byers, M. Dixon, W.D. Smart, and C.M. Grimm, "Say cheese!: Experiences with a robot photographer," in *Proc. Fifteenth Innovative Applications of Artificial Intelligence Conf. (IAAI-03)*, Acapulco, Mexico, August 2003.
- [3] J. Barreto, P. Menezes, and J. Dias, "Human-robot interaction based on haar-like features and eigenfaces," in *Proc. 2004 IEEE Int. Conf. on Robotic and Automation*, New Orleans, LA, USA, 2004, pp. 1888–1893.
- [4] S. Lang, M. Kleinhagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer, "Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot," in *Proc. Int. Conf. on Multimodal Interfaces*, Vancouver, Canada, 2003, pp. 28–35.
- [5] *Proceedings of the 1st IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, Grenoble, France, March 2000.
- [6] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proc. Int. Conf. on Pattern Recognition*, Barcelona, Spain, September 2000, pp. 4167–4170.
- [7] C. J. Needham and R. D. Boyle, "Performance evaluation metrics and statistics for positional tracker evaluation," in *Proc. Int. Conf. on Computer Vision Systems*, Graz, Austria, April 2003, pp. 278–289.
- [8] J. Black, T. J. Ellis, and P. Rosin, "A novel method for video tracking performance evaluation," in *Proc. Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, October 2003, pp. 125–132.
- [9] C. E. Erdem, A. M. Tekalp, and B. Sankur, "Metrics for performance evaluation of video object segmentation and tracking without ground truth," in *Proc. Int. Conf. on Image Processing*, Thessaloniki, Greece, October 2001.
- [10] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Proc. European Conf. on Computer Vision*, Prague, Czech Republic, May 2004.
- [11] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.
- [12] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *Int. Journal of Computer Vision*, vol. 29(1), pp. 5–28, 1998.