Person Tracking based on a Hybrid Neural Probabilistic Model

Wenjie Yan, Cornelius Weber, and Stefan Wermter

University of Hamburg, Department of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany
{yan,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. This article presents a novel approach for a real-time person tracking system based on particle filters that use different visual streams. Due to the difficulty of detecting a person from a top view, a new architecture is presented that integrates different vision streams by means of a Sigma-Pi network. A short-term memory mechanism enhances the tracking robustness. Experimental results show that robust real-time person tracking can be achieved.

Keywords: person detection, particle filter, neural network, multimodality

1 Introduction

Artificial neural networks (ANN) are widely used to model complex behavior and are applied in different fields, such as computer vision, pattern recognition, and classification. They can also be used to overcome the major challenge of real-time person tracking in a complex ambient intelligent environment. In this paper we present a novel approach of indoor person tracking using a single ceiling-mounted camera with a fish-eye lens.

A few person tracking systems based on ceiling mounted cameras have been proposed previously [6],[12]. However, it is hard to get a robust tracking ability based on a single feature. A person observed from the top view produces very different shapes at different locations thus it is difficult to be recognized by fixed patterns. Motion provides a good tracking indicator but cannot provide information when a person does not move. The color obtained from the clothes can be a reliable tracking feature, but we have to learn the color information first from other information. Different vision information in combination, however, can be used to detect and localize a person's position reliably.

A hybrid knowledge-based architecture tackles the specific challenges that arise from this setup by integrating different vision streams into a Sigma-Pi network [13]. A person can be localized using a particle filter based on the output of this network. The system architecture and the used methods are presented in section 2 and 3. The experimental results are shown in section 4. A discussion is presented in section 5 and section 6 concludes this article.

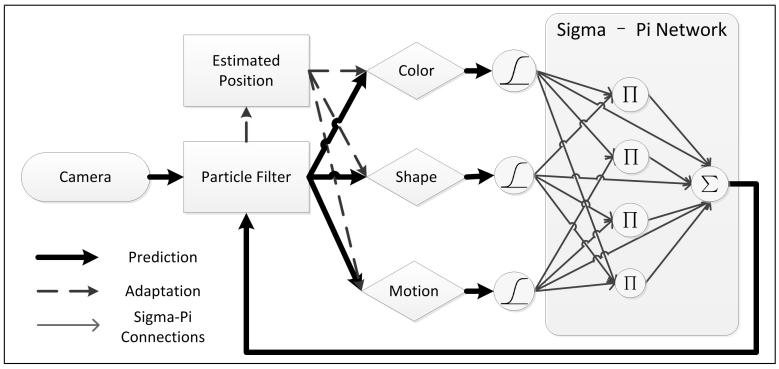

Fig. 1. System architecture

2 Methods

Our model is illustrated in Figure 1. A Sigma-Pi network integrates shape, motion and color streams and passes its output to a particle filter, which provides robust object tracking based on the history of previous observations [10]. The work flow can be split into two parts: *prediction* and *adaptation*. In the prediction phase (see arrows in Figure 1), each particle segments a small image patch and evaluates this patch using three visual cues. The activities of visual cues are generated via activation functions and scaled by their connection weights, which are called reliabilities here. Through the Sigma-Pi network, the weights of particles are computed and then the position of the particles will be updated. In the adaptation phase, the reliability weights of the Sigma-Pi network will be adapted. The estimated position of the person will be validated again using the visual cues (see arrows for adaptation) and weights will be calculated based on the validation results. With the collaborative contribution of each cue, the tracking performance can be improved significantly.

2.1 Particle Filter

Particle filters are an approximation method that represents a probability distribution with a set of particles and weight values. A particle filter is usually integrated in partially observable Markov decision processes (POMDPs) [5]. A POMDP model consists of unobserved states of an agent s, in our case the position of the observed person, and observations of the agent z. A transition model $P(s_t|s_{t-1})$ describes the probability that the state changes from s_{t-1} to s_t at time t. If the agent executes the action a_{t-1}, $P(s_t|s_{t-1}, a_{t-1})$ can be estimated based on the transition model. For simplicity, let us assume here that we do not know anything about the person's actions. Based on the Bayesian formulation, the agent's state can be estimated according to an iterative equation:

$$P(s_t|z_{0:t}) = \eta P(z_t|s_t) \int P(s_{t-1}|z_{0:t-1}) P(s_t|s_{t-1}) \mathrm{d}s_{t-1} \tag{1}$$

where η is a normalization constant, $P(z_t|s_t)$ is the observation model and $P(s_t|z_{0:t})$ is the probability of a state given all previous observations from time 0 to t. In a discrete model, the probability of the state s_t can be computed recursively from the previous distribution $P(s_{t-1}|z_{0:t-1})$:

$$P(s_t|z_{0:t}) \approx \eta P(z_t|s_t) \sum_i \pi_{t-1}^{(i)} P(s_t|s_{t-1}^{(i)}) \tag{2}$$

In the particle filter, the probability distribution can be approximated with a set of particles i in the following form:

$$P(s_t|z_{0:t}) \approx \sum_i \pi_{t-1}^{(i)} \delta(s_t - s_{t-1}^{(i)}) \tag{3}$$

where $\pi^{(i)}$ denotes the weight factor of each particle with $\sum \pi^{(i)} = 1$ and δ denotes the Dirac impulse function. The mean value of the distribution can be computed as $\sum_i \pi_{t-1}^{(i)} s_t$ and may be used to estimate the state of the agent if the distribution is unimodal.

At the beginning of the tracking, the particles are placed randomly in the image. Then a small patch surrounding them is taken and probed to detect the person with the visual cues. Where the sum of weighted cues returns large saliencies, the particles will get larger weight values, raising the probability of this particle in the distribution and showing that a person is more likely to be in this position. In order to keep the network exploring, 5% particles are assigned to random positions in each step.

2.2 Sigma-Pi Network

In the tracking system, the weight factor $\pi^{(i)}$ of particle i will be computed with a Sigma-Pi network [13]. The activities of the different visual cues are set as the input of the Sigma-Pi network and the weights are calculated with the following equation:

$$\begin{aligned} \pi^{(i)} = & \sum_c^3 \alpha_c^l(t) A_c(s_{t-1}^{(i)}) + \sum_{c_1 > c_2}^3 \alpha_{c_1 c_2}^q(t) A_{c_1}(s_{t-1}^{(i)}) A_{c_2}(s_{t-1}^{(i)}) \\ & + \alpha_{c_3}^c(t) A_{c_1}(s_{t-1}^{(i)}) A_{c_2}(s_{t-1}^{(i)}) A_{c_3}(s_{t-1}^{(i)}) \end{aligned} \tag{4}$$

where $A_c(s_{t-1}^{(i)}) \in [0, 1]$ is the activity of cue c at the position of particle i, which can be thought of as taken from a saliency map over the entire image [4]. The network weights $\alpha_c^l(t)$ denote the linear reliability and $\alpha_{c_1 c_2}^q(t)$ and $\alpha_{c_3}^c(t)$ are the quadratic and cubic combination reliabilities of the different visual cues. Compared with traditional multi-layer networks, the Sigma-Pi network contains the correlation and higher-order correlation information between the input values. The reliability of some cues, like motion, are non-adaptive, while others, like color, need to be adapted on a short time scale. This requires a mixed adaptive framework, as inspired by models of combining different information [11], [2]. An

issue is that an adaptive cue will be initially unreliable, but when learned may have a high quality in predicting the person's position. To balance the changing qualities between the different cues, the reliabilities will be evaluated with the following equation:

$$\alpha(t) = (1 - \epsilon)\alpha(t-1) + \epsilon f(s'_t) + \beta \tag{5}$$

where ϵ is a constant learning rate and β is a constant. $f(s'_t)$ denotes an evaluation function and is computed by the combination of visual cues' activities:

$$f_c(s'_t) = \sum_{i \neq c}^{n} A_i(s'_t) A_c(s'_t) \tag{6}$$

where s'_t is the estimated position and n is the number of the reliabilities. In this model n is 7 and contains 3 linear and 4 combination reliabilities. The function is large when more cues are active at the same time, which leads to an increase of the cue's reliability α.

3 Processing Different Visual Cues

The image patches segmented by the particles are evaluated by the visual cues. Three independent cues, *motion*, *shape* and *color* are used in this model to extract different features from the image.

Motion detection is a method to detect an object by measuring the difference between images. We use here the background subtraction method [7] that compares the actual image with a reference image. Since the background stays mostly constant, the person can be found when the difference of image is larger than a predefined threshold. Considering that the background may also change, as when furniture is being moved, the background is updated smoothly using a running average. When the new input image remains static for a longer time, for example a person sits in a chair, the background will be converted to the new image, the person will merge into the background and then he will not be detected anymore. In this case, the shape and color cue will allow the system to find the person.

Color is an important feature for representing an object. Because the color of objects and people does not change quickly, it is a reliable feature for tracking. The image is converted to the HSV color space to reduce the computation effort [8]. Using a histogram backprojection algorithm [9], a saliency value image is generated that shows the probability of the pixels of the input image that belong to the example histogram. For each particle, the pixel values of the probability image inside of the segmentation window are accumulated. The higher the value is, the more this image segment matches the histogram pattern.

Since *shape* contains information irrelevant of the light condition as well as the surface texture, it represents significant features of an object. We extract here SURF features [1] for describing the image objects. Because the shape of

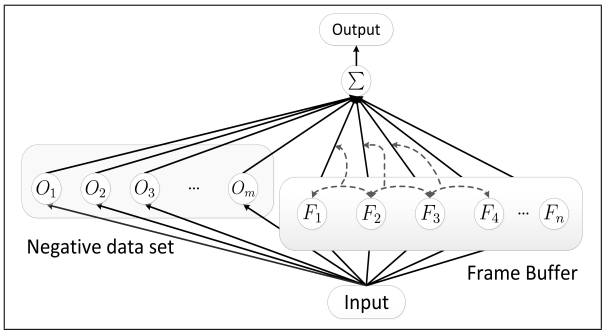

Fig. 2. Shape cue

a person from the top view changes significantly and is hard to be described by static patterns, a short time memory mechanism is conceived to track the person based on previous features. A feature buffer stores the image features of the last 10 frames. The correlations between the new input image feature and the features in the 10 frames are calculated. Based on these values, the output activity is calculated via an activation function. If the change of the person's shape is continuous and slow, the features of neighboring frames in the buffer should be similar. Weights of the buffer images are calculated using the matching rates between the adjacent frames. Features from a negative background data set such as sofas, tables and chairs have a negative contribution to the shape cue, which helps the particles to avoid the background.

4 Experimental Results

The environment for testing the tracking system is shown in Figure 3. The camera image is calibrated and subsampled to the resolution 320×240, which allows real-time processing. Image material from 6 videos have been tested. The experiment aims to detect and locate a person or a mobile robot under static condition in the image as well as to track their motion trajectories when moving. One person will be tracked in the experiment. Different image noises, for example when changing the furniture's position, changing the person's appearance and also disturbance by another person are tested. 50 particles were used for the person tracking and therefore only a small part of the images is being processed. This accelerates the system in comparison with a search window method.

4.1 Tracking a Person

The mechanism of the person tracking system is demonstrated in Figure 3. The particles are initialized at random positions in the image. When a person enters the room, the weight values of the nearby particles will increase so that the particles move towards the person. The shape feature as well as the color histogram

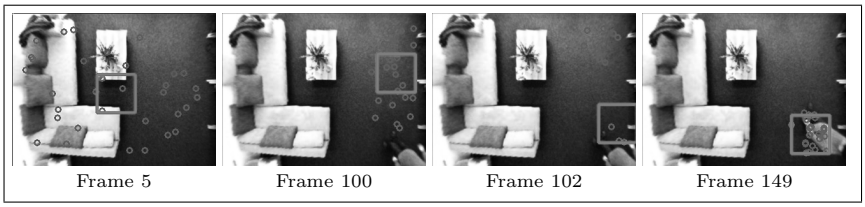

Fig. 3. Tracking a person moving into the room

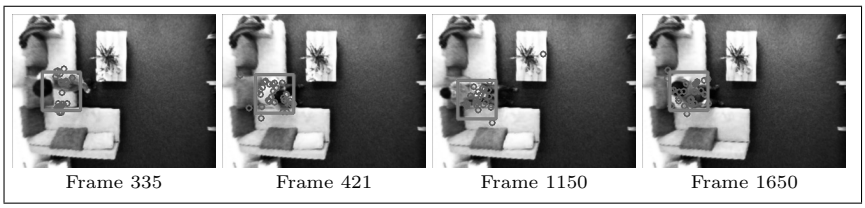

Fig. 4. Tracking a sitting person

will adapt themselves at the same time. When a person does not move, when sitting as in Figure 4, the motion cue is missed but then the shape and color memory will recover the system to detect the person.

4.2 Changing Environment

The disturbance of a changing environment, for example a moving table in the room (Figure 5) will automatically be corrected by the negative feedback of the shape cue. Although the particles may follow the motion cue, the shape of the table from the background model returns a negative feedback to the shape cue, which helps the particles to go back to the person. The experimental results are

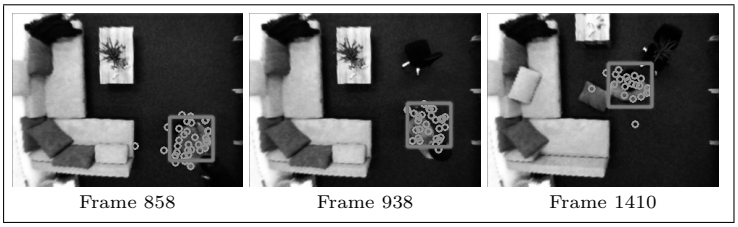

Fig. 5. Person tracking during change of environment

summarized in Table 1. 90.28% of the images on average are tracked correctly. In comparison, the success rate of tracking a person based on single motion detection could reach only 69% on average.

Table 1. Experiment results

Name	Total Frames	Missing	Mismatch	Success rate (%)
Person Moving 1	2012	19	22	97.96
Person Moving 2	2258	169	12	91.98
Person Moving and Sitting 1	1190	78	21	91.68
Person Moving and Sitting 2	980	22	130	84.18
Change Environment 1	1151	89	30	89.66
Change Environment and Distracter Person 1	1564	157	141	80.94
Total	9155	534	356	90.28

5 Discussion

We have presented a new hybrid neural probabilistic model that adapts its behavior online based on different visual cues. The model is to some extent indicative of a human's ability of recognizing objects based on different features. When some of the features are strongly distorted, detection recovers by the integration of other features. The particle filter parallels an active attention selection mechanism, which allocates most processing resources to positions of interest. It has a high performance of detecting complex objects that move relatively slowly in real time. Accordingly, our model has potential as a robust method for object detection and recognition in complex conditions. It may in the future be better if the system tracks a person not only based on these three cues, but also on some further features. Also, non-visual sensors could be used such as a microphone, which provides auditory data to enhance the tracking accuracy.

The short-term memory enables the system to localize objects rapidly without a-priori knowledge about the target person. We have experimented with a multilayer perceptron network based on moment invariant features [3] that was trained to recognize a person. However, due to the variety of the person's shape observed from the top view and its similarity to the furniture, this method was not efficient to distinguish the person from the background. Nevertheless, we are considering to include a person-specific cue in the future.

6 Conclusions

In this paper we have presented a novel approach for real-time person tracking based on a ceiling-mounted camera. A hybrid probabilistic algorithm is proposed for localizing the person based on different visual cues. A Sigma-Pi architecture integrates the output of different cues together with corresponding reliability factors. Advantages of this system are that the feature pattern used for one cue, such as the color histogram, can adapt on-line to provide a more robust identification of a person. With this short-term memory mechanism, the system processes images from an unstructured environment as well as moving objects in

a real ambient intelligent system. We are planning to generalize this architecture with a recurrent memory neural network and improve the quality of visual cues to obtain higher tracking precision and extend the functions for detecting the pose of a person.

Acknowledgements This research has been partially supported by the KSERA project funded by the European Commission under the 7th Framework Programme (FP7) for Research and Technological Development under grant agreement n°2010-248085, and the EU project RobotDoc under 235065 ROBOT-DOC from the 7th Framework Programme, Marie Curie Action ITN.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision - ECCV 2006, LNCS, vol. 3951, pp. 404 – 417. Springer Berlin / Heidelberg (2006)
2. Bernardin, K., Gehrig, T., Stiefelhagen, R.: Multi-level particle filter fusion of features and cues for audio-visual person tracking. In: Multimodal Technologies for Perception of Humans, LNCS, vol. 4625, pp. 70 – 81. Springer Berlin / Heidelberg (2008)
3. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8(2), 179 – 187 (1962)
4. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11), 1254 – 1259 (Nov 1998)
5. Kaelbling, L., Littman, M., Cassandra, A.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101(1-2), 99–134 (1998)
6. Nait-Charif, H., McKenna, S.: Activity summarisation and fall detection in a supportive home environment. In: Proceedings of the 17th International Conference on Pattern Recognition. vol. 4, pp. 323 – 326 (2004)
7. Piccardi, M.: Background subtraction techniques: a review. In: IEEE International Conference on Systems, Man and Cybernetics. vol. 4, pp. 3099 – 3104 (2004)
8. Sural, S., Qian, G., Pramanik, S.: Segmentation and histogram generation using the hsv color space for image retrieval. In: International Conference on Image Processing. vol. 2, pp. 589 – 592 (2002)
9. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7, 11–32 (1991)
10. Thrun, S.: Particle filters in robotics. In: Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI). vol. 1 (2002)
11. Triesch, J., Malsburg, C.: Democratic integration: Self-organized integration of adaptive cues. Neural Computation 13(9), 2049 – 2074 (2001)
12. West, G., Newman, C., Greenhill, S.: From Smart Homes to Smart Care, chap. Using a Camera to Implement Virtual Sensors in a Smart House, pp. 83 – 90 (2005)
13. Zhang, B., Muhlenbein, H.: Synthesis of Sigma-Pi neural networks by the breeder genetic programming. In: Proceedings of the First IEEE Conference on Evolutionary Computation. vol. 1, pp. 318 – 323 (Jun 1994)