Journal of Ambient Intelligence and Smart Environments 3 (2011) 237–252
DOI 10.3233/AIS-2011-0111
IOS Press

A hybrid probabilistic neural model for person tracking based on a ceiling-mounted camera

Wenjie Yan [*], Cornelius Weber and Stefan Wermter
University of Hamburg, Department of Informatics, Knowledge Technology, Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany

Abstract. Person tracking is an important topic in ambient living systems as well as in computer vision. In particular, detecting a person from a ceiling-mounted camera is a challenge since the person's appearance is very different from the top or from the side view, and the shape of the person changes significantly when moving around the room. This article presents a novel approach for a real-time person tracking system based on particle filters with input from different visual streams. A new architecture is developed that integrates different vision streams by means of a Sigma-Pi-like network. Moreover, a short-term memory mechanism is modeled to enhance the robustness of the tracking system. Based on this architecture, the system can start localizing a person with several cues and learn the features of other cues online. The experimental results show that robust real-time person tracking can be achieved.

Keywords: Person detection, person recognition, particle filter, neural network

1. Introduction

Ambient Intelligence (AmI) refers to environments equipped with sensitive, intelligent devices that react to motion or other signals of a person and support their life [4]. An AmI environment system is able to monitor a person using a ubiquitous sensor system and to assist with life activities by means of actuators. In particular, Ambient Assisted Living (AAL) addresses the caretaking of elderly people and patients and is regarded as one of the most important fields in AmI [4,39]. According to the estimate of the U.S. Census Bureau, the population aged over 65 will grow from 13% to 20% from 2010 to 2030 [19] due to worldwide population aging. In Europe, more than 20% of the population will be beyond 60 by 2020 [52] and by 2050 even 37% will be beyond 60 [2]. It is expected that there will be a large gap of persons at working age to support the persons at the age of 65 or older. Hence, this increase motivates the development of autonomous, intelligent home care systems.

In an AmI environment, different sensors are installed to gather personal information. After data analysis in the AmI server, the status of a person can be estimated and the AmI system can provide appropriate help and predict emergencies which then may be avoided by means of warnings. A reliable person localization functionality is essential for the AmI system to ensure that the status estimation is correct.

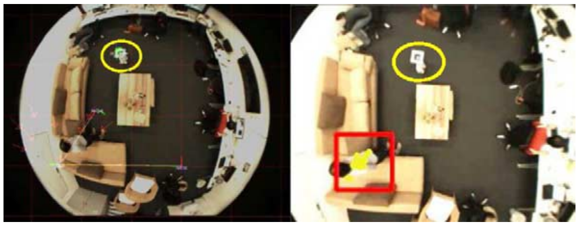


Fig. 1. Robot navigation based on person and robot localization by a ceiling-mounted camera. The red bounding box indicates the position of a target person and the yellow oval denotes the position of a robot.

[*] Corresponding author. E-mail: yan@informatik.uni-hamburg.de.

With the development of technologies, the importance of human-machine interaction in an AmI system increases steadily [10,34]. In the absence of a care giver, a service robot can assist a person's life by bringing medication, providing useful information, displaying videos using a portable beamer or supporting a video communication. A robust person tracking system is therefore important and enables the robot to navigate to the person's position (Fig. 1).

However, person tracking in a complex home environment is a major challenge for AmI systems. Non-vision-based techniques of person localization, such as those based on RFID tags [32,46] or radio waves, require the person to carry certain technical devices which are unsuitable in everyday situations. Motion sensors [6,60] can detect a person entering or leaving a room, but cannot provide the precise location information. Infrared cameras are costly and suffer from high degrees of noise in indoor settings. Compared to these approaches, a vision system promises to provide good performance and a wide use scope at a reasonable cost. The vision system provides far more information than the other kinds of sensors. It can be assessed whether the person is standing, sitting or moving, as well as an emergency situation such as a fall [38]. Hence, for tracking a person, a color camera is our main sensor of choice. Privacy concerns of camera surveillance can be addressed by not storing image information, if the person's location is needed only for a short time.

In general, it is hard to get robust visual tracking ability in a real, complex and unpredictable home environment based on visual input. For example, a person observed from the top produces very different shapes at different locations thus it is difficult to be recognized by static patterns (Fig. 2). A motion detector may provide a good tracking indicator but cannot provide information when a person does not move, for example when the person is sitting on a sofa. The situation could also be disturbed by moving environments and changing light conditions. The color obtained from the clothes and skin can be a reliable tracking feature, but in a real life scenario, we have to learn the color information first from other information since the color of a person's clothes can change every day. Multiple camera systems can help the tracking ability, but these systems are expensive, complex and hard to install.

Considering that different visual information sources can be used in combination to detect and localize a person's position reliably, a hybrid knowledge-based architecture is approached that integrates the different visual streams into a Sigma-Pi network architecture [61]. The system is able to start localizing a person with some of the cues and adapts the other cues online. The reliabilities of cues, which indicate the importance of each cue for decision making, are also adapted. Based on the output of this network, a particle filter [13] updates the probability distribution to estimate the person's position. A single ceiling-mounted camera with a fish-eye lens is used to keep the system simple and easy to install.

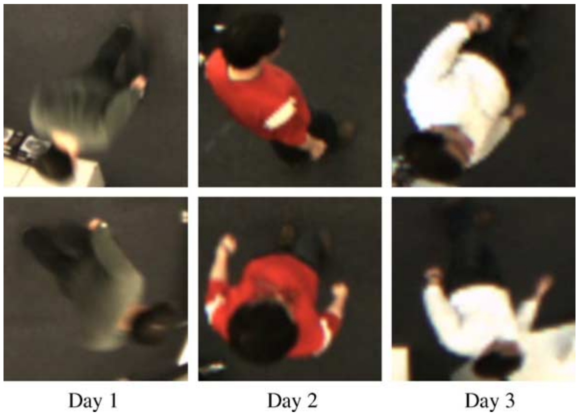


Fig. 2. Person images from a ceiling-mounted camera. It is hard to define a person by a fixed shape pattern.

The context and related works are introduced in Section 2. The system architecture is presented in Section 3 and each visual cue will be described in Section 4. The experimental results will be shown in Section 5 with an evaluation and Section 6 concludes this article with a discussion.

2. Context and related work

A growing number of research groups are developing AmI systems. An early approach was developed in 1999 by a research group at the Massachusetts Institute of Technology and industry partners in the Oxygen project to create an environment that is aware of people's needs without requiring people learning how to use computers [47]. The Aware Home of Georgia Institute of Technology aims at developing a smart home that is aware of its own state and the state of its inhabitants [30]. Industrial companies, such as Philips [3] and Sony [1] have been committed to the development of AmI system for many years.

In recent years, the human-machine interaction has been drawing more attention to the AmI systems. The ambient environment will not only react passively to a change of a situation, but will also provide active help, such as via home electronics, motorized fittings or – in the future – service robots. In the EU-funded project Knowledgeable Service Robots for Aging (KSERA) we are developing a socially assistive robot that supports some activities of daily life as well as health care needs of an elderly person, specifically persons suffering from Chronic Obstructive Pulmonary Disease. For this purpose, the status of the person will be observed by sensors and anticipated with the help of statistical or neural prediction. A small humanoid robot Nao [15] is the main actuator that delivers feedback from the AAL system to the person. For example, it gives health advice based on medical sensor readings and acts as a mobile communication platform for the person with remote care givers.

In many AAL settings, persons are detected indirectly for instance by measuring the open-closed state of doors and drawers, or via passive infrared sensors [56]. The precision of localization based on such status information is very low, while on the other hand, laser and stereo vision [50] offer high precision at a high cost. Other suggested additional devices are motion sensors worn by the tracked person [7], using correlation of the motion sensor's signal with the motion registered by the camera. Person tracking based on multiple sensors [31,40] can obtain extra information, but the system complexity arises due to the data fusion and system configuration.

Person tracking based on vision is a very active research area. For instance, stereo vision systems [5,37] can use the 3D information reconstructed by different cameras to distinguish easily a person from the background. Multiple ceiling-mounted cameras are used in combination [49] to compensate for the narrow field-of-view of a single camera [33], or to overcome shadowing and occlusion problems [26]. Although these multi-camera systems can detect and track multiple persons, they are expensive and complex. For example, the camera system has to be calibrated carefully not only to eliminate the distortion effect of lens, but also to indicate the correlation between different cameras.

A single ceiling-mounted camera is another possibility for person tracking. West et al. [59] have developed a ceiling-mounted camera model in a kitchen scenario to infer interaction of a person with kitchen devices. The single ceiling-mounted camera can be calibrated easily or can be used even without calibration. With a wide-angle view lens, for example a fisheye lens, the ceiling-mounted camera can observe the entire room. Moreover, occlusion is not a problem since the camera is at a position to see a person at any position within the room.[1] The main disadvantage of the single ceiling-mounted camera setup is the limited raw information contained by the camera. Therefore, a sophisticated algorithm is essential to track a person.

There are many person detection methods based on computer vision. The most common technique for detecting a moving person is background subtraction [43], which finds the person based on the difference between an input and a reference image. Appearance-based models have been researched in the recent years. Principal component analysis (PCA) [24] and independent component analysis (ICA) [22], for instance, represent the original data in a low dimensional space by keeping major information. Some other methods like scale-invariant feature transformation (SIFT) [35] or a speeded up robust feature (SURF) [8] detect interest points (for example using Harris corner [17]) for object detection. These methods are scale- and rotation invariant and are able to detect similarities in different images. However, the computation complexity of these methods is high and they perform poorly with non-rigid objects. Person tracking based on body part analysis [14,18,45] can detect a person precisely, but requires a very clear body shape captured from a front view. In this case, a multiple camera system has to be installed in a room environment to always get the body shape. The color obtained from the clothes and skin can be a reliable tracking feature [11,37,62], but this may have to be adapted quickly after when the color changes.

In our approach we use a single ceiling-mounted camera to track a person. Different visual information is combined to detect and localize a person's position reliably, inspired by a model of combining different information for face tracking [55]. Our approach can track a person with or without motion information, and is robust against environment noise such as moving furniture, changing light conditions and interacting with other people. The target person can be memorized through the adaptivity of the cues which act as a memory and enable the system to select a specific person for tracking. A particle filter approach [28,33,44,51], which has potential for tracking, is developed to localize the person based on visual cues, that are being adaptively combined in a Sigma-Pi network architecture.

[1] Only the small humanoid Nao robot, which we also use in our project can sometimes be hidden by furniture, as seen from the ceiling camera, but knowledge about its motor commands and odometry allows for approximate position estimation from history.

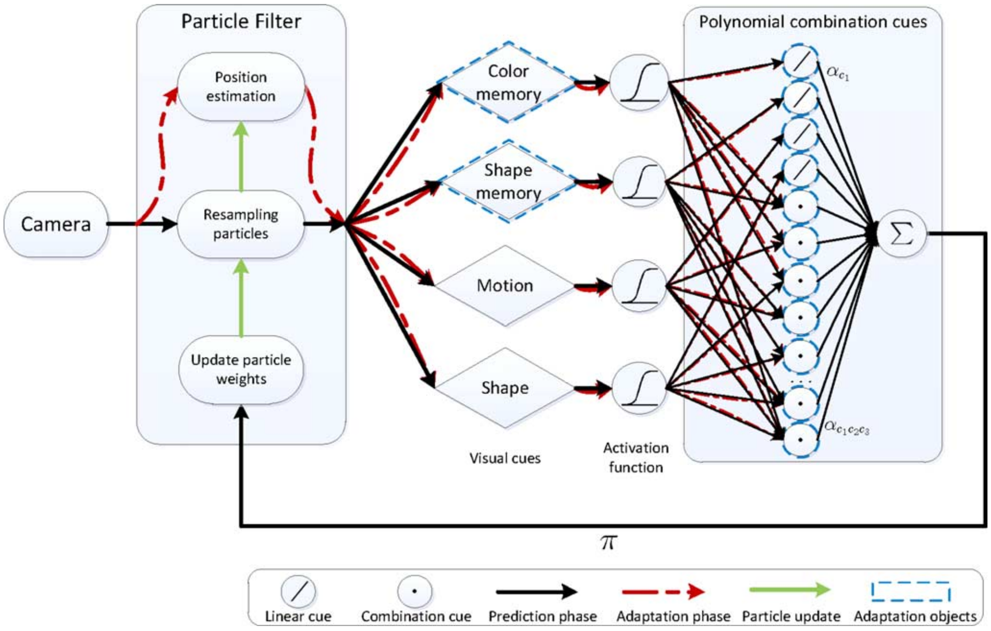


Fig. 3. Architecture of the tracking system.

3. System overview

In our home scenario, a person and a small humanoid robot are being tracked. While both of them are being tracked simultaneously, for simplicity, we will only describe the algorithm for person tracking. Human and humanoid tracking differ only in the training data used for the neural network that analyses the tracked object's shape.

Our model is illustrated in Fig. 3. A Sigma-Pi network architecture integrates shape, motion, color memory and shape memory streams and passes its output to a particle filter which provides robust object tracking based on the history of previous observations [13,20,57,58]. The work flow can be split into two parts: *prediction* and *adaptation*.

In the prediction phase (black arrows in Fig. 3), each particle segments a small image patch and evaluates this patch using the visual cues. Four cues are used: a color memory cue based on the histogram, a motion cue based on background subtraction, a fixed shape cue based on a neural network and a shape memory cue based on SURF features. The activities of visual cues are generated via activation functions and scaled by their connection weights which are called reliabilities here. Through the polynomial combination of cues represented by a Sigma-Pi network, the weights of particles are computed. The particles will then be resampled and the position of the particles will be updated (green arrows). After that, 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 dashed red arrows in Fig. 3). The color memory cue and shape memory cue will be learned and the reliabilities of visual cues, which will be described in Section 3.2, will be adapted based on the validation results (labeled with the blue dashed lines). With the collaborative contribution of each cue, the tracking performance can be improved significantly.

3.1. *Particle filters*

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) [25]. 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}, the further $P(s_t|s_{t-1}, a_{t-1})$ can be estimated based on the transition model. For simplicity, let us assume here that we do not know about a person's actions.

Algorithm 1 Sequential Importance Resampling (SIR)

```
Draw samples for $N$ particles from the proposal distribution:
    $s_t^{(i)} \sim q(s_t) = \sum_i \pi_{t-1}^{(i)} P(s_t|s_{t-1}^{(i)})$
Update the importance weight $\pi_t^{(j)}$:
    $\pi_t^{(j)} = \pi_{t-1}^{(j)} P(z_t|s_t^{(j)})$
Normalize the importance weights $\{\pi_t^{(j)}\}$:
    $\pi_t^{(j)} = \frac{\pi_t^{(j)}}{\sum_k \pi_t^{(k)}}$
Compute the effective number of particles:
    $\hat{N}_{\text{eff}} = \frac{1}{\sum_{j=1}^{N} \left(\pi_t^{(j)}\right)^2}$
If $\hat{N}_{\text{eff}}$ is less than a threshold, resample the particles with the probabilities proportional to their weights and reset the weight values:
    $s_t^{(j)} \propto \pi_t^{(j)}$
    $\pi_t^{(j)} = \frac{1}{N}$,    for    $j = 1 \ldots N$
```

Based on the Bayesian representation in a POMDP, the agent's state can be estimated as:

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

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. Because $P(s_t|z_{0:t})$ describes "what the state looks like", it is also called the belief of the state.

In a discrete computing model, the belief of the state s_t at time t under the observation $z_{0:t}$ can be computed recursively according to the previous distribution $P(s_{t-1}|z_{0:t-1})$:

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

where the probability distribution of the states is represented with a set of particles $\{i\}$, with each particle i containing the state information. The beliefs of the states are expressed by corresponding weight values $\pi^{(i)}$. Hence, the probability distribution can be approximated in the form:

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

where π denotes the weight factor of each particle with $\sum \pi = 1$ and δ denotes the Dirac impulse function. The higher the weight value, the more important this particle is in the whole distribution. 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.

There are different ways to model a particle filter, and we use the sequential importance resampling algorithm which is described in Algorithm 1. In the person tracking system, the person's position is represented by the x- and y-coordinates in the image, i.e. $s = \{x, y\}$. The direction of a person's motion is hard to predict, because, for example, an arm movement during rest could be wrongly perceived as a body movement into the corresponding direction. Hence, we do not use direction of movement information, but describe the transition model $P(s_t|s_{t-1}^{(i)}, a_{t-1})$ of the person with a Gaussian distribution:

$$P(s_t|s_{t-1}^{(i)}, a_{t-1}) = \frac{1}{\sqrt{2\pi\sigma(a)^2}} e^{-\frac{(s_{t-1}^{(i)} - s_t^{(i)})^2}{2\sigma(a)^2}} \quad (4)$$

where $\sigma(a)^2$ is the variance, $s_{t-1}^{(i)}$ are the previous states, $s_t^{(i)}$ is the current states and a_{t-1} is the executed action. Movement information from the motion cue (see Section 4.2) in the action variable a_t, however, is informative for the person's movement distribution which we account for by increasing $\sigma(a)$ when motion is detected. The $\sigma(a)$ is then set to either of two values:

$$\sigma(a) = \begin{cases} v_1 & \text{if motion detected} \\ v_2 & \text{else} \end{cases} \quad (5)$$

where v_1, v_2 are constant parameters with $v_1 > v_2$. When no motion is detected, the probabilistic distribution will shrink to a small area that allows the particles only to move close to the previous position. This modulates the behavior in a way that when an object is identified, a human would remember its position when the object does not move.

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 replaced with random positions at each step to search for possible position of a person actively. This strategy accelerates the system much compared with traditional pixelwise search window methods.

3.2. Sigma-Pi network

In the tracking system, the weight factor $\pi^{(i)}$ of particle i will be computed with a weighted polynomial combination of visual cues inspired by the Sigma-Pi network [61]. The activities of the different visual cues are set as the input of the Sigma-Pi network and the particle weights are calculated with the following equation:

$$\begin{aligned}\pi^{(i)} = & \sum_{c}^{4} \alpha_c^l(t) A_c(s_{t-1}^{(i)}) \\ & + \sum_{c_1>c_2}^{4} \alpha_{c_1c_2}^q(t) A_{c_1}(s_{t-1}^{(i)}) A_{c_2}(s_{t-1}^{(i)}) \\ & + \sum_{c_1>c_2>c_3}^{4} \alpha_{c_1c_2c_3}^c(t) A_{c_1}(s_{t-1}^{(i)}) \\ & A_{c_2}(s_{t-1}^{(i)}) A_{c_3}(s_{t-1}^{(i)}) \end{aligned} \tag{6}$$

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 [23]. The activities are scaled by a sigmoid activation function which can be described with Eq. (7):

$$A(y_c) = \frac{1}{1 + e^{-(g \cdot y_c)}} \tag{7}$$

where y_c is the output of the visual cues and g is a constant scale factor. The coefficients of the polynomial cues, i.e. the network weights $\alpha_c^l(t)$ denote the linear reliability, $\alpha_{c_1c_2}^q(t)$ and $\alpha_{c_1c_2c_3}^c(t)$ are the quadratic and cubic combination reliabilities of the different visual cues. The quadratic and cubic combinations of the four basic cues yield the further combination 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 timescale. This requires a mixed adaptive framework, as inspired by models of combining different information [9,55]. An issue is that an adaptive cue will be initially unreliable, but when adapted it 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:

$$\alpha(t) = (1-\epsilon)\alpha(t-1) + \epsilon f(s_t') + \beta \tag{8}$$

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

$$f_c(s_t') = \sum_{i \neq c}^{n} A_i(s_t') A_c(s_t') \tag{9}$$

where s_t' is the estimated position and n is the number of the reliabilities. In this model n is 14 and contains 4 linear, 6 quadratic and 4 cubic combination reliabilities. The function is large when more cues are active at the same time, which leads to an increase of the cues' reliability α. The details of each visual cue will be introduced in the next section.

4. Processing different visual cues

The *motion cue*, *shape cue*, *shape memory cue* and *color memory cue* are used to extract features from the image and to update the probability of the person or the robot at the particle's position. For the *shape cue*, we use the moment invariants to present the shape information and train a multilayer perceptron (MLP) network to classify the input image patch [29]. An MLP network has been chosen based on its robust classification learning properties. For the *motion cue*, a background subtraction method has been implemented. For the *color memory cue*, the probability of image areas that belong to the estimated image position is computed using a histogram backprojection algorithm [54]. The *shape memory cue* is based on a set of SURF features [8] weighted by the correlation with adjacent frames. The details of these methods are introduced in the following paragraphs.

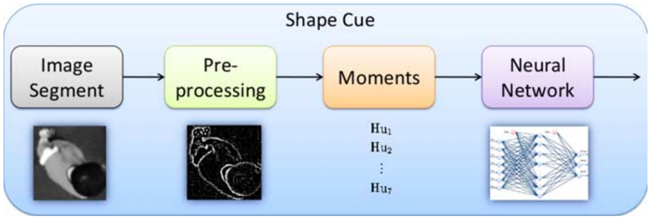


Fig. 4. Processing of the shape cue.

4.1. Shape cue

Since shape contains information irrelevant of the light condition as well as the surface texture, it is usually used to present the significant feature of the object in the image classification tasks. As shown in Fig. 4, the image patch of the particles is preprocessed by a Laplace filter with 3 × 3 pixel kernel and converted to a counter image (see Fig. 4). The moment invariant features are extracted based on these counter images and used as input to a multilayer perceptron neural network. We collected the training data of person, Nao robot and background noise. The data collection contained 6000 images. 75 percent of the data are used for learning and 25 percent for testing. After the training phase, the network is able to classify new images. For each particle, we take the output node "person" of the neural network as the shape cue value.

4.1.1. Moment invariants

In computer vision, moment invariants are essential to analyze or recognize objects independent of their position, rotation or scale [29,36]. Because the person's shape, i.e. the counter image is converted using a Laplace filter, changes significantly when moving within the ceiling-mounted camera's sight, it is difficult to detect the person using common pattern matching methods. The Hu-Moments [21] provide therefore a good method to solve this problem. For a $M \times M$ grey-value image, the two-dimensional moments can be computed as follows:

$$M_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} x^p y^q f(x,y);\ p,q = 0,1,2,\dots \quad (10)$$

where x, y are the positions of image pixels and $f(x,y)$ is the intensity of point (x,y). Moments can represent features of the image, for example the first order moments can be used to calculate the center of the mass $(\bar{x},\bar{y})$ with:

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \text{and} \quad \bar{y} = \frac{M_{01}}{M_{00}} \quad (11)$$

which includes also the central moments. It is also possible to recalculate complex moments based on the raw moments. A moment translated by (a,b) can be represented as:

$$\mu_{pq} = \sum_x \sum_y (x+a)^p (y+b)^q f(x,y) \quad (12)$$

The central moment μ_{pq} can be described as:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \quad (13)$$

Normalizing the central moment with μ_{00}, we get the scale invariant moments using the following equation:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\left(1+\frac{p+q}{2}\right)}} \quad (14)$$

According to the invariant moments, seven scale, position and orientation invariant moments can be calculated with the following equations:

$$\begin{aligned}
M_1 &= (\eta_{20} + \eta_{02}) \\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} + \eta_{03})^2) + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad (3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\
M_6 &= (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
M_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} + \eta_{03})^2) \\
&\quad - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03}) \\
&\quad (3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \qquad (15)
\end{aligned}$$

This set of values is also called the Hu-Moments.

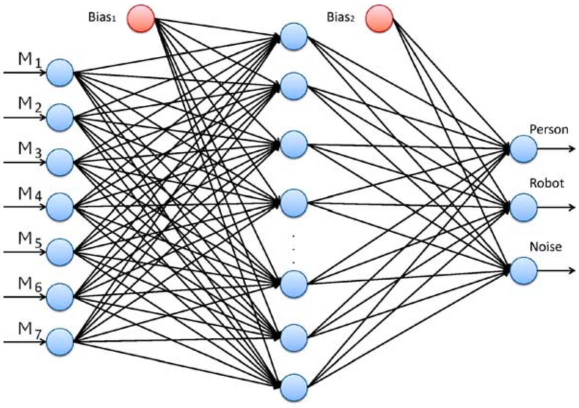


Fig. 5. MLP network for shape classification with 7 Hu-Moments as inputs (see Eq. (15)).

4.1.2. Multilayer perceptron

To detect the person from the moment invariants, we train a multilayer perceptron [41,48] network for the classification. This artificial neural network consists of multiple layers of neurons which are connected fully with the neurons in the neighbor layers. An MLP network can be used for function approximation and classification based on supervised learning. The MLP for shape classification is shown in Fig. 5. Seven input neurons connect directly with the Hu-Moments. In the middle layer we use 30 neurons with the sigmoid activation function.

There are three output neurons which represent the detection of person, robot and noise. We train and test the neural network with 3 groups of training images (*person*, *robot* and *noise*), each of them containing 1500 images for learning and 500 for testing. The back-propagation algorithm is used to train the network. For each training step, an error value between the desired output and the actual output of the MLP is computed:

$$E = \frac{1}{2}\sum_i (y_i^{\text{out}} - d_i)^2 \tag{16}$$

where y_i^{out} is the output of the neuron i in the output layer and d_i is the desired output. A learning rule is applied to update the weight value of each connection weight in the network:

$$w(t) = w(t-1) + \Delta w(t) \tag{17}$$

with

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \tag{18}$$

After the training phase, the neural network is able to generalize and classify new images. The output of the neural network can be intuitively interpreted as "whether the image segment looks like a person or a robot". The output neuron of the person returns the classification result, which means $y_c^{\text{shape}} = y_{\text{person}}^{\text{out}}$.

If it is the same as the group index of particle filters, for instance the "person" output of the MLP network is high and this particle belongs also to the group "person", the shape cue will receive a strong feedback. The reliability $\alpha_s(t)$ of the shape cue will be updated according to Eq. (8).

4.2. Motion cue

Motion detection is a method to detect an object by measuring difference in the image. We use here the background subtraction method [43] 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. We convert the image from RGB color space to the grey value color space. The intensity is subtracted from the reference image using:

$$M(\mathbf{x}, t) = |i(\mathbf{x}, t) - i'(\mathbf{x}, t-1)| \tag{19}$$

where $i(\mathbf{x}, t)$ is the intensity of image $\mathbf{x}$ at the time t and $i'(\mathbf{x}, t-1)$ is the intensity of the reference image at time $t-1$. The difference $M(\mathbf{x}, t)$ is compared with a threshold h and the area containing motion is defined using a step function:

$$f(M(\mathbf{x}, t)) = \begin{cases} 0, & \text{if } M(\mathbf{x}, t) - h \leqslant 0 \\ 1, & \text{if } M(\mathbf{x}, t) - h > 0 \end{cases} \tag{20}$$

The pixels are merged with blob detection which allows that the connected pixels are labeled with the same blob index and the motion objects are segmented with this method. Compared with a reference size of a person, the motion cue returns the likelihood y_c^{motion} that a moving object is a person as:

$$y_c^{\text{motion}} = e^{-\frac{(s_{\text{mo}} - s_{\text{ref}})^2}{2c^2}} \tag{21}$$

where c denotes a fixed variance, s_{mo} is the size of the motion object and s_{ref} is the reference size of a person.

Considering that the background may also change, as when moving the furniture, the background is updated smoothly with the following formula:

$$i'(\mathbf{x},t) = (1-\gamma)i'(\mathbf{x},t-1) + \gamma i(\mathbf{x},t) \qquad (22)$$

where $\gamma \ll 1$ is an update rate. When the new input image remains static for a longer time, for example while a person is sitting in a chair, the background will be converted to the new image, the person will merge into the background and then will not be detected anymore. In this case, the other visual cues will allow the system to find the person.

4.3. Color memory cue

Color is an important feature for representing an object, for example the cloth color of a person and the surface color of an object. Since the color of objects and person does not change quickly, it is a reliable feature for tracking.

A large number of tracking methods are based on color information [12,42,62]. A histogram is used here to describe the tracking target. A histogram in computer vision is a representation of the color distribution in an image. Since the HSV color space is more efficient for a computer vision system than the RGB color space, the image colors are converted to the HSV space [53]. Because the color information is mainly represented by the Hue value (in RGB space the color information is distributed in three dimensions), we use a one-dimensional histogram to represent the Hue information.

Using a histogram backprojection algorithm [54], a gray value image is generated that shows the probability of the pixels of the input image that belong to the example histogram. The histogram backprojection method computes the ratio histogram R_i according to the target histogram O_i and the histogram of new input image I_i:

$$R_i = \min\left(\frac{O_i}{I_i}, 1\right) \qquad (23)$$

where i denotes the index of bins in the histogram. The target histogram O_i is updated according to the evaluation of shape and color cues. When the evaluation receives a positive feedback, the target histogram will update with the following formula:

$$O_i(t) = (1-\zeta)O_i(t-1) + \zeta I_i(t) \qquad (24)$$

where ζ denotes an update rate. The ratio histogram R_i represents the probability that a color belongs to the target image. The pixel value of the new input image will be replaced with the corresponding value R_i considering the color index. For each particle, the pixel values of the probability image inside of the segmentation window are accumulated and return y_c^{color}. The higher the value is, the more this image segment matches the histogram pattern.

Considering that the tracked person might wear clothes with different colors at different days, there is no defined color pattern for tracking at the beginning and the cue of the color model is thought of as unreliable. However, when the correct color pattern is found, the color matching model will be reliable because the clothes' color of a person does not change on a short time scale. Hence, the dynamic cue adaptation should help the shape classification to dominate the person recognition when the color matching or the motion detection are missing, and support the color cue for decision making when the color information is learned.

4.4. Shape memory cue

The shape memory cue is based on the target person found in the previous frames. Because the status of a person is continuous, a short time memory mechanism has been developed to track the person based on previous features. We extract SURF features [8] for representing the image objects. A feature buffer stores the image features of the last 30 frames. The correlations between the new input image feature and the features in the previous frames are calculated. Considering that 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 avoid the background. The system structure is shown in Fig. 6. The output of this network returns the shape memory cue y_c^{sm}.

5. Experiments and results

The environment for testing the tracking system is shown in Fig. 7. The fish-eye lens is calibrated and the camera image is subsampled to the resolution 320 × 240, which allows real-time processing. 10 different videos have been tested. The experiment aims to detect and locate a person or a mobile robot under static conditions in the image as well as to track their motion

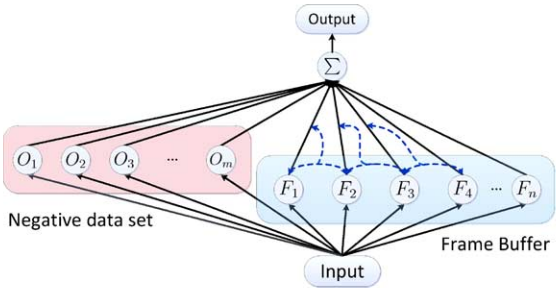


Fig. 6. Structure of the shape memory cue.

trajectories when moving. One person will be tracked in the experiment. Different disturbances, for example when changing the furniture's position, changing the person's appearance and the disturbance by another person are tested. 30 particles were used for the person tracking and therefore only a small part of the images is being processed. The segment area of the particle filters are set as 60×60 pixels according to the approximate size of the tracked object. This accelerates the system in comparison with a search window method.

A reference image is captured at the beginning to obtain the initial background model. The SURF features of furniture in the background model are stored as negative data set. When a person moves in the room without a planned color pattern, the shape and motion cues will detect the person and the particles will merge to the position of the person. The histogram of the estimated position of the person will be updated and the SURF features of this image patch will be extracted and be pushed into the memory buffer. The reliabilities of visual cues will be adapted according to Eq. (8).

Different experimental scenarios were designed as follows:

- Tracking a moving person
- Tracking a sitting person
- Changing light condition
- Changing furniture position
- Distracter person
- Distracter person using CLEAR 07 data set [27]

All these tests have been carried out in our ambient laboratory. A similar room or data set can also be used to evaluate the algorithm, but the calibration of the new room and the training data of the MLP network are needed. The detail description of scenarios as well as their results are shown in the following sections.

5.1. *Tracking a person moving in the room*

A test example is shown in Fig. 7. The motion cue will facilitate finding the person in this test. At the beginning (frame 5), the particles are initialized at random positions in the image. When a person enters the room (frame 100), the weight values of the nearby particles will increase so that the particles move towards the person. The person will be detected and localized quickly (frame 149). The shape feature as well as the color histogram will adapt themselves at the same time.

5.2. *Tracking a sitting person*

A person can stay in a position for a long time, for example while watching television, reading a book, etc. In this case no motion will be detected and the motion cue will temporarily not work. It is important to test if the other visual cues can help the system to continue the tracking successfully. A person moves to the sofa and sits down in our experiment and the particles can keep localizing the person for a long time.

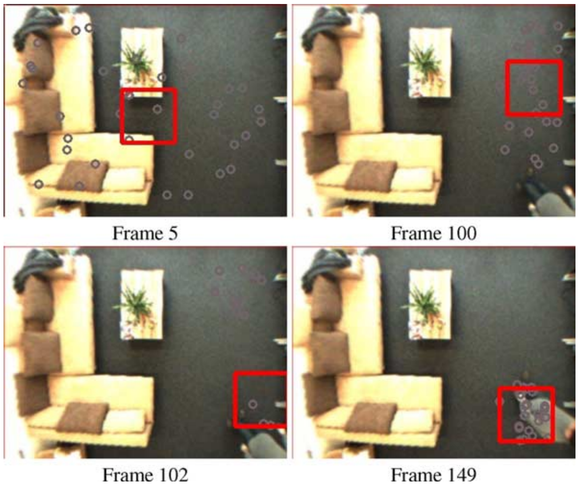
5.3. *Changing light condition*

In a real environment, the light condition changes continuously. It causes a problem for person tracking because of the modified features and it is a large challenge for map building as well as for robot navigation. In this task we challenge the person localization by changing the light condition. After a person is located by the particle filters, we switch on/off some of the lights.

One setup is shown in Fig. 8. We switch the lights off and on after the person is localized by the particle filter (frame 85). Due to the dramatic change of the intensity, the particles lose the target person (frame 105). But after a short time they recover and return to the person (frame 115).

5.4. *Changing furniture position*

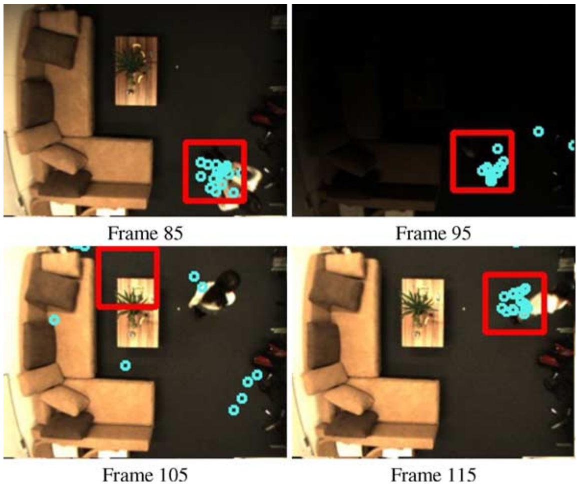
Another challenge is to modify the room structure. The disturbance of a changing environment, for example a moving table in the room (Fig. 9) will automatically be corrected by the negative feedback of the shape cue. Although the particles may follow the motion cue, the shape of the table from the background model returns a negative feedback to the shape cue, which helps the particles go back to the person soon.



Frame 5 Frame 100

Frame 102 Frame 149

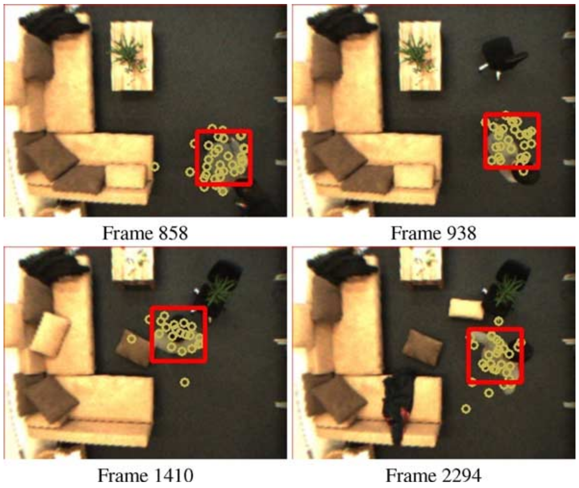
Fig. 7. Tracking a person moving into the room.



Frame 85 Frame 95

Frame 105 Frame 115

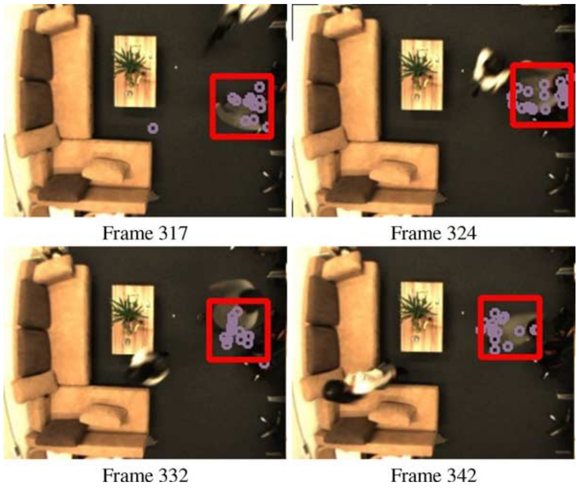
Fig. 8. Changing light condition.



Frame 858 Frame 938

Frame 1410 Frame 2294

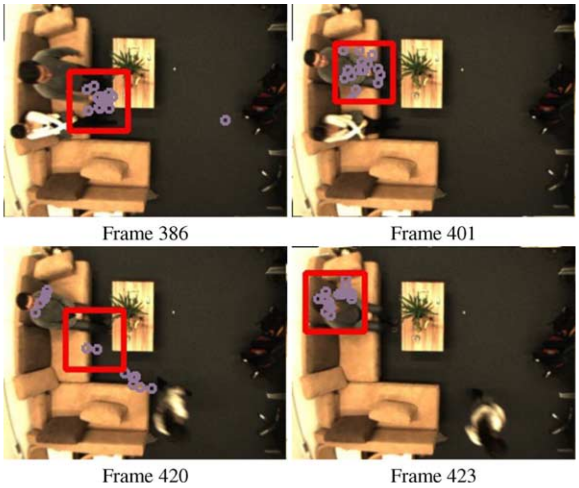
Fig. 9. Person tracking during change of environment.



Frame 317 Frame 324

Frame 332 Frame 342

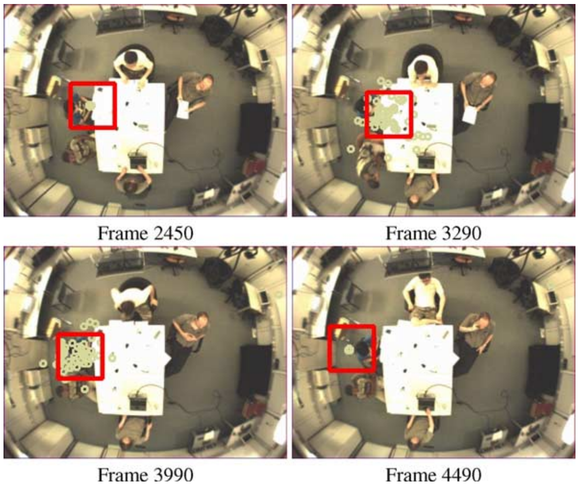
Fig. 10. A person crossing.



Frame 386 Frame 401

Frame 420 Frame 423

Fig. 11. Person sitting close on a sofa.



Frame 2450 Frame 3290

Frame 3990 Frame 4490

Fig. 12. Test with CLEAR 07 data set.

Table 1

Experiment results

Name	Total Frame	m	fp	MOTA (%)
Person moving scenario 1	2012	19	22	97.96
Person moving scenario 2	2258	169	12	91.98
Person moving and sitting scenario 1	1190	78	21	91.68
Person moving and sitting scenario 2	980	22	130	84.18
Change environment scenario 1	1151	89	30	89.66
Change environment scenario 2	1564	157	141	80.94
Change light condition in night scenario	160	17	59	52.5
Change light condition in day scenario	540	0	3	99.45
Distracter person scenario 1	1014	48	35	91.81
Distracter person scenario 2	700	57	26	88.14
Distracter person scenario CLEAR 07	2122	188	52	88.68
Total	13691	844	531	89.96

5.5. *Distracter person*

The target of presented tracking system is to localize a single person, but it is common that multiple persons are in the room. To select a specific person among them for tracking is therefore essential for the system. In this task we test the possibility of tracking a target person when another person is in the room. Two persons will move in the room, sit on the sofa together and move again. The memory cue and the learned color cue will recover the system when being disturbed by the motion of the other person.

In Fig. 10 we show a test scenario that two persons walk across. A person is tracked at the test beginning (frame 317). When two people come very close (frame 324) to each other, the particles are still able to keep tracking the target person. Figure 11 shows another test scenario. The target person sits first on the sofa close to another person (frame 386). Since the target person does not move, the motion cue is disabled (frame 401). After that, when the other person stands up and moves, the particles are disturbed strongly by the motion cue (frame 420). But the color and memory cue will recover the system quickly and the particles will come back to the target person again (frame 423).

5.6. *Distracter person using CLEAR 07 data set*

We conducted a set of experiments based on the fish-eye camera video of CLEAR 07 short sample data set to evaluate tracking performance based on external data. The idea of our system is to monitor a target person when being alone in the room. Because the CLEAR 07 multiple person tracking data set aims to track multiple persons, our current system will rely on selecting a target person. Therefore, we can only evaluate the system when one of the persons is tracked. The experiment is shown in Fig. 12. When a person is tracked successfully, the person will always be localized until the end of this video.

5.7. *Evaluation*

The experimental results have been evaluated principally according to the CLEAR MOT Metrics [27]. Since only a single person is tracked in the system, based on our goal design, the frame number of misses m and of false positives fp has been counted and the multiple object tracking accuracy (MOTA) has been calculated. The threshold distance of a false positive was defined as 40 pixels; 11 videos were evaluated and the results are summarized in Table 1. We can see that 89.96% of the images on average are tracked correctly. The best case is the change light condition in the day scenario which indicates that the slight change of light under sufficient sunshine does not disturb the tracking system at all. The worst case is the change light condition in the night scenario. However, it is also the hardest test because the lamps are the only light source. The light condition is changed totally when most of the lamps are switched off and a person can hardly be observed from the camera video (see frame 95 in Fig. 8). In comparison, the success rate of tracking person based on single motion detection could reach 69% on average and the color and memory cue alone can not achieve the tracking task.

5.7.1. *Reliabilities*

The contribution of visual cues can be evaluated by their reliability values. The more often a visual cue

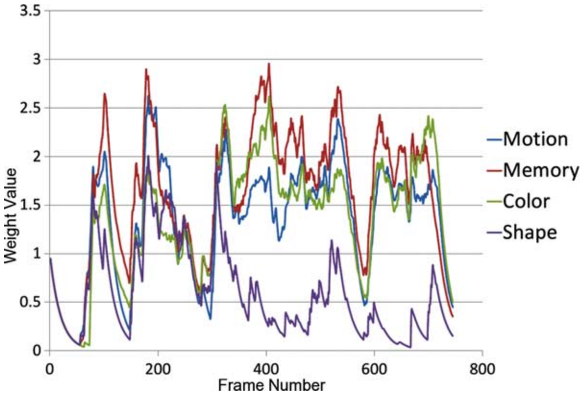


Fig. 13. Reliabilities of linear cues.

helps to find the person, the higher the reliability of this cue will get. The reliabilities of linear visual cues are shown in Fig. 13. The x axis of the diagram denotes the frame number and the y axis the weight values. At the beginning of the tracking, the color cue has a small value since the histogram has not yet been learned. When the color information is trained (for example after frame 300), the color cue arises to a high value that makes it important for the system. The memory cue has usually a high value because this cue memorizes the shape of the target person which is very reliable. The motion cue has a median importance but it is essential to notice the other cues. The shape cue has a lower value than the others, because the shape of a person changes always and is hard to be classified continually. Nevertheless, this shape cue does help the system to find the person in the image at the beginning.

5.7.2. *Computational complexity*

The worst case of computation complexity of the used visual cues is $\mathcal{O}(n^2)$, where n denotes the width of the quadratic search window. Because these visual cues are computed for each particle, the cost of computing visual cues of all particles is then $\mathcal{O}(pmn^2)$, where m denotes the number of particles and p the number of linear visual cues. To assign the reliabilities it takes $\mathcal{O}(p^2)$ and to resample the particles costs $\mathcal{O}(m)$. Thus the total computational effort is $\mathcal{O}(pmn^2) + \mathcal{O}(p^2m) + \mathcal{O}(m)$. Because the number of visual cues is constant, for example here $p = 4$, the total cost is then $\mathcal{O}(mn^2)$. The particle filter accelerates here the system speed in comparison with a pixelwise search window method, because only a few particles (30) process a small part of the images (60×60 pixels) at each step. In Table 2 we list the computation time for 100 frames with different number of particles. The particles are able to track the person correctly throughout all these tests. Therefore this system is shown to work under real-time condition.

Table 2

Computational time with different particle numbers

Particles numbers	Frames	Used time (*ms*)
30	100	2947
50	100	4525
100	100	8063
200	100	15385
500	100	31266
1000	100	61483

6. Conclusion and future work

In this paper we have presented a novel approach for real-time detecting and tracking a person from a ceiling-mounted camera view. A hybrid probabilistic algorithm is proposed for localizing the person based on different visual cues. A Sigma-Pi like network integrates the output of different cues together with corresponding reliability factors which helps a particle filter to track the person. 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 disturbed, 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.

Advantages of this system are that the feature pattern used for one cue, such as the color histogram, can adapt online to provide a more robust identification of a person. With this short-term memory mechanism, the system could master the challenge of an unstructured environment as well as moving objects in a real ambient intelligent system. Accordingly, our model has potential as a robust method for object detection and tracking in complex conditions. We are planning to equip 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 the person.

6.1. *Future work*

It may in the future be better if the system tracks a person not only based on these four cues, but also on some further features. Principally, the more cues there are, the better tracking performance we could get. In addition, non-visual sensors could be used such as a microphone, which provides new data to improve 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 [21] that was trained to recognize a person. However, due to the variety of the person's shape observed from the top view, this a-priori knowledge about the person can be improved to distinguish the person from the background. We are considering to include another person-specific cue in the future.

Though the initial design of the tracking system is to monitor a single person when the person is alone at home, it might be interesting to extend the system to track multiple persons. Through the experiments our system has been shown to track a specific person while other people can be in the room (see Section 5.5). Our hybrid system has the potential to achieve multiple people tracking as well. The particle filter framework will be adapted for multiple person tracking, for example using the RJMCMC algorithm [16].

Acknowledgments

The research leading to these results is part of the KSERA project (http://www.ksera-project.eu) 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] Sony interaction laboratory. http://www.sonycsl.co.jp/IL/. Date accessed: 30. November, 2010.

[2] OECD demographic and labour force database, Technical report, Organisation for economic co-operation and development, 2007.

[3] E. Aarts and B. Eggen, Ambient Intelligence in Homelab, Philips Research, 2002.

[4] E. Aarts, R. Harwig, and M. Schuurmans, *Ambient Intelligence, The Invisible Future: The Seamless Integration of Technology into Everyday Life*, McGraw-Hill, Inc., New York, NY, 2001.

[5] S. Bahadori, L. Iocchi, G. Leone, D. Nardi, and L. Scozzafava, Real-time people localization and tracking through fixed stereo vision, *Applied Intelligence* **26** (2007), 83–97. 10.1007/s10489-006-0013-3.

[6] T.S. Barger, D.E. Brown, and M. Alwan, Health-status monitoring through analysis of behavioral patterns, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **35**(1) (January 2005), 22–27.

[7] G. Bauer and P. Lukowicz, Developing a sub room level indoor location system for wide scale deployment in assisted living systems, in: *Computers Helping People with Special Needs*, K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, eds, Lecture Notes in Computer Science, Vol. 5105, Springer, Berlin / Heidelberg, 2008, pp. 1057–1064. 10.1007/978-3-540-70540-6_158.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, Surf: Speeded up robust features, in: *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, eds, Lecture Notes in Computer Science, Vol. 3951, Springer, Berlin / Heidelberg, 2006, pp. 404–417. 10.1007/11744023_32.

[9] K. Bernardin, T. Gehrig, and R. Stiefelhagen, Multi-level particle filter fusion of features and cues for audio-visual person tracking, in: *Multimodal Technologies for Perception of Humans*, R. Stiefelhagen, R. Bowers, and J. Fiscus, eds, Lecture Notes in Computer Science, Vol. 4625, Springer, Berlin / Heidelberg, 2008, pp. 70–81. 10.1007/978-3-540-68585-2_5.

[10] M. Broxvall, M. Gritti, A. Saffiotti, B.S. Seo, and Y.J. Cho, PEIS Ecology: integrating robots into smart environments, in: *Proc. of the IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, May 2006, pp. 212–218.

[11] D. Comaniciu, V. Ramesh, and P. Meer, Real-time tracking of non-rigid objects using mean shift, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, Los Alamitos, CA, USA, IEEE Computer Society, 2000, pp. 2142–2150.

[12] J. Czyz, B. Ristic, and B. Macq, A particle filter for joint detection and tracking of color objects, *Image and Vision Computing* **25**(8) (2007), 1271–1281.

[13] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, Particle filters for mobile robot localization, in: *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, eds, Springer-Verlag, 2001, pp. 499–516.

[14] S. Frintrop, A. Königs, F. Hoeller, and D. Schulz, A component-based approach to visual person tracking from a mobile platform, *International Journal of Social Robotics* **2** (2010), 53–62. 10.1007/s12369-009-0035-1.

[15] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, Mechatronic design of nao humanoid, in: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, May 2009, pp. 769–774.

[16] P.J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika* **82**(4) (1995), 711.

[17] C. Harris and M. Stephens, A combined corner and edge detector, in: *Alvey Vision Conference*, Vol. 15, Manchester, UK, 1988, p. 50.

[18] F. Hecht, P. Azad, and R. Dillmann, Markerless human motion tracking with a flexible model and appearance learning, in: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, May 2009, pp. 3173–3179.

[19] J.M. Hootman and C.G. Helmick, Projections of US prevalence of arthritis and associated activity limitations, *Arthritis & Rheumatism* **54**(1) (2006), 226–229.

[20] A. Howard, Multi-robot simultaneous localization and mapping using particle filters, *The International Journal of Robotics Research* **25**(12) (2006), 1243.

[21] M.K. Hu, Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory* **8**(2) (February 2002), 179–187.

[22] A. Hyvärinen and E. Oja, Independent component analysis: Algorithms and applications, *Neural Networks* **13**(4–5) (2000), 411–430.

[23] L. Itti, C. Koch, and E. Niebur, A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11) (November 1998), 1254–1259.

[24] I. Jolliffe, *Principal Component Analysis*, John Wiley & Sons, Ltd, 2005.

[25] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, Planning and acting in partially observable stochastic domains, *Artificial Intelligence* **101**(1–2) (1998), 99–134.

[26] K. Kemmotsu, Y. Koketsua, and M. Iehara, Human behavior recognition using unconscious cameras and a visible robot in a network robot system, *Robotics and Autonomous Systems* **56**(10) (2008), 857–864.

[27] B. Keni and S. Rainer, Evaluating multiple object tracking performance: The CLEAR MOT metrics, *EURASIP Journal on Image and Video Processing* **2008** (2008), 10.

[28] S.M. Khan and M. Shah, Tracking multiple occluding people by localizing on multiple scene planes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31** (2009), 505–519.

[29] A. Khotanzad and J.H. Lu, Classification of invariant image representations using a neural network, *IEEE Transactions on Acoustics, Speech and Signal Processing* **38**(6) (June 2002), 1028–1038.

[30] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter, The aware home: A living laboratory for ubiquitous computing research, in: *Cooperative Buildings. Integrating Information, Organizations, and Architecture*, N. Streitz, J. Siegel, V. Hartkopf, and S. Konomi, eds, Lecture Notes in Computer Science, Vol. 1670, Springer, Berlin / Heidelberg, 1999, pp. 191–198. 10.1007/10705432_17.

[31] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, People tracking and following with mobile robot using an omnidirectional camera and a laser, in: *Proc. of the IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, May 2006, pp. 557–562.

[32] J. Koch, J. Wettach, E. Bloch, and K. Berns, Indoor localisation of humans, objects, and mobile robots with RFID infrastructure, in: *7th International Conference on Hybrid Intelligent Systems, 2007. HIS 2007*, Vol. 0, Los Alamitos, CA, USA, IEEE Computer Society, 2007, pp. 271–276.

[33] O. Lanz and R. Brunelli, An appearance-based particle filter for visual tracking in smart rooms, in: *Multimodal Technologies for Perception of Humans*, R. Stiefelhagen, R. Bowers, and J. Fiscus, eds, Lecture Notes in Computer Science, Vol. 4625, Springer, Berlin / Heidelberg, 2008, pp. 57–69. 10.1007/978-3-540-68585-2_4.

[34] A. Louloudi, A. Mosallam, N. Marturi, P. Janse, and V. Hernandez, Integration of the humanoid robot nao inside a smart home: A case study, in: *The Swedish AI Society Workshop*, Uppsala University, May 2010. http://www.ep.liu.se/ecp/048/008/.

[35] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60** (2004), 91–110. 10.1023/B:VISI.0000029664.99615.94.

[36] M. Mercimek, K. Gulez, and T.V. Mumcu, Real object recognition using moment invariants, *Sadhana* **30** (2005), 765–775. 10.1007/BF02716709.

[37] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente, People detection and tracking using stereo vision and color, *Image and Vision Computing* **25**(6) (2007), 995–1007.

[38] H. Nait-Charif and S.J. McKenna, Activity summarisation and fall detection in a supportive home environment, in: *Proc. of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, Vol. 4, August 2004, pp. 323–326.

[39] J. Nehmer, M. Becker, A. Karshmer, and R. Lamm, Living assistance systems: an ambient intelligence approach, in: *Proc. of the 28th International Conference on Software Engineering*, ICSE '06, New York, NY, USA, ACM, 2006, pp. 43–50.

[40] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, A joint particle filter for audio-visual speaker tracking, in: *Proc. of the 7th International Conference on Multimodal Interfaces*, ICMI '05, New York, NY, USA, ACM, 2005, pp. 61–68.

[41] S.K. Pal and S. Mitra, Multilayer perceptron, fuzzy sets, and classification, *IEEE Transactions on Neural Networks* **3**(5) (September 1992), 683–697.

[42] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, Color-based probabilistic tracking, in: *Computer Vision – ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, eds, Lecture Notes in Computer Science, Vol. 2350, Springer, Berlin / Heidelberg, 2002, pp. 661–675. 10.1007/3-540-47969-4_44.

[43] M. Piccardi, Background subtraction techniques: A review, in: *2004 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, 2004, pp. 3099–3104.

[44] Y. Qian, G. Medioni, and I. Cohen, Multiple target tracking using spatio-temporal markov chain Monte Carlo data association, in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07*, June 2007, pp. 1–8.

[45] D. Ramanan, D.A. Forsyth, and A. Zisserman, Tracking people by learning their appearance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29** (2007), 65–81.

[46] Y. Raoui, M. Goller, M. Devy, T. Kerscher, J.M. Zollner, R. Dillmann, and A. Coustou, RFID-based topological and metrical self-localization in a structured environment, in: *International Conference on Advanced Robotics, ICAR 2009*, 2009, pp. 1–6.

[47] L. Rudolph, Project oxygen: Pervasive, human-centric computing – an initial experience, in: *Advanced Information Systems Engineering*, K. Dittrich, A. Geppert, and M. Norrie, eds, Lecture Notes in Computer Science, Vol. 2068, Springer, Berlin / Heidelberg, 2001, pp. 1–12. 10.1007/3-540-45341-5_1.

[48] D.E. Rumelhart, G.E. Hintont, and R.J. Williams, Learning representations by back-propagating errors, *Nature* **323**(6088) (1986), 533–536.

[49] A. Salah, R. Morros, J. Luque, C. Segura, J. Hernando, O. Ambekar, B. Schouten, and E. Pauwels, Multimodal identification and localization of users in a smart environment, *Journal on*

Multimodal User Interfaces **2** (2008), 75–91. 10.1007/s12193-008-0008-y.

[50] B. Schiele, M. Andriluka, N. Majer, S. Roth, and C. Wojek, Visual people detection – different models, comparison and discussion, in: *Proc. of the ICRA 2009 Workshop on People Detection and Tracking*, K.O. Arras and O. Martinez Mozos, eds, 2009.

[51] K. Smith, D. Gatica-Perez, and J.M. Odobez, Using particles to track varying numbers of interacting people, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1** (2005), 962–969.

[52] H. Steg, H. Strese, C. Loroff, J. Hull, and S. Schmidt, Europe is facing a demographic challenge Ambient Assisted Living offers solutions, IST Project Report on Ambient Assisted Living, March 2006.

[53] S. Sural, G. Qian, and S. Pramanik, Segmentation and histogram generation using the hsv color space for image retrieval, in: *Proc. of the International Conference on Image Processing*, Vol. 2, 2002, pp. II-589–II-592.

[54] M.J. Swain and D.H. Ballard, Color indexing, *International Journal of Computer Vision* **7** (1991), 11–32. 10.1007/BF00130487.

[55] J. Triesch and C. Malsburg, Democratic integration: Self-organized integration of adaptive cues, *Neural Computation* **13**(9) (2001), 2049–2074.

[56] T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse, Activity recognition using semi-Markov models on real world smart home datasets, *Journal of Ambient Intelligence and Smart Environments* **2** (2010), 311–325.

[57] C. Weber and S. Wermter, A self-organizing map of sigma-pi units, *Neurocomputing* **70**(13–15) (2007), 2552–2560.

[58] S. Wermter, G. Palm, and M. Elshaw, *Biomimetic Neural Learning for Intelligent Robots: Intelligent Systems, Cognitive Robotics, and Neuroscience*, Lecture Notes in Computer Science, Vol. 3575, Springer, Heidelberg, 2005.

[59] G. West, C. Newman, and S. Greenhill, Using a camera to implement virtual sensors in a smart house, in: *From Smart Homes to Smart Care*, IOS Press, 2005, pp. 83–90.

[60] A.Y. Yang, R. Jafari, S.S. Sastry, and R. Bajcsy, Distributed recognition of human actions using wearable motion sensor networks, *Journal of Ambient Intelligence and Smart Environments* **1**(2) (2009), 103–115.

[61] B. Zhang and H. Muhlenbein, Synthesis of sigma-pi neural networks by the breeder genetic programming, in: *Proc. of the First IEEE Conference on Evolutionary Computation*, June 1994, pp. 318–323.

[62] Z. Zivkovic and B. Krose, An EM-like algorithm for color-histogram-based object tracking, in: *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, Vol. 1, June 2004, pp. I-798–I-803.