ICT-2009.7.1
KSERA Project
2010-248085

Deliverable D2.2

Context Awareness for Navigation

16 March 2011
Public Document

The KSERA project (http://www.ksera-project.eu) has received funding from the European Commission under the 7th Framework Programme (FP7) for Research and Technological Development under grant agreement n°2010-248085.

Project acronym:	KSERA
Project full title:	Knowledgeable SErvice Robots for Aging
Work package:	2
Document number:	D2.2
Document title:	Context Awareness for Navigation
Version:	1
Delivery date:	30 March 2010 (month 14)
Actual publication date:	15.04.2011
Dissemination level:	Public
Nature:	Report

Editor(s) / lead beneficiary:	Stefan Wermter (UH)
Authors(s):	Cornelius Weber (UH), Wenjie Yan (UH), David van der Pol (TU/e), Elena Torta (TU/e), Raymond Cuijpers (TU/e)
Reviewer(s):	Junpei Zhong (UH), Nils Meins (UH)

Contents

Glossary of terms used

Acronym	Definition
HHI	Human Human Interaction
HRI	Human Robot Interaction
Naoqi	Client-server software architecture for programming Nao
Nao	Humanoid robot manufactured by Aldebaran Robotics
Paro	Therapeutic robotic baby seal
Kismet	Robotic head developed at MIT
REC	Region of Eye Contact
Aldebaran Robotics	Manufacturer of the Nao robot
AAL	Ambient Assisted Living
AmI	Ambient Intelligence
COPD	Chronic Obstructive Pulmonary Disease
KSERA	Knowledgeable Service Robots for Aging
PS	Personal space

Executive summary

This deliverable describes research progress of person/robot localization and the robot navigation model. In the KSERA project, we are developing a socially assistive robot that supports some activities of daily life as well as health care needs of an elderly person, especially persons suffering from Chronic Obstructive Pulmonary Disease (COPD). A small humanoid robot Nao is designed as the main actuator that delivers feedback from the AAL system to the person and takes care of the person with remote care givers. The robot and people localization and the robot navigation ability are therefore essential to achieve the task. We introduce here our real-time person tracking system based on particle filters with different visual streams. Our novel architecture integrates different vision streams by means of a Sigma-Pi-like network. Moreover, a short-term memory mechanism is modelled to enhance the robustness of the tracking system. A behavior-based robot navigation architecture is implemented that allows the robot to safely navigate in a cluttered and dynamically changing domestic environment, encodes embodied non-verbal interactions: the robot respects the user's personal space (PS) by choosing the appropriate distance and direction of approach. The experimental results show that robust real-time robot/people localization and robot navigation can be achieved.

Purpose of this deliverable

In this deliverable (D2.2) we report our current progress on context awareness navigation. The research described in this report contributes to the person's and robot's localization in an ambient intelligence environment and the robot navigation based on the people's localization. The main focus of the research has been to model, implement and test a robust people localization algorithm under different conditions and to test the people-based robot navigation.

Suggested readers

This document is recommended to all KSERA partners. The research is of interest to the scientific community as well as industry developing socially assistive robots. Any research or development in which the robot should cooperate with a human being can benefit form the research presented in this report.

Relationship to other documents

The context awareness and navigation abilities are required for the prototypes of robot mobile behaviour described in deliverables D2.3 and D2.4 and for the integrated prototypes described in the coming deliverables of WP 5. This document is also the basis for the coming deliverable D2.5, which reports on navigation based on context awareness and intention reading.

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 [1]. An AmI environment system is able to monitor the person using a ubiquitous sensor system and to assist with daily life activities by means of actuators. In particular, Ambient Assisted Living (AAL) addresses the care-taking of elderly people and patients and is regarded as one of the most important fields in AmI [1, 2]. 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 [3] due to worldwide population aging. In Europe, more than 20% of the population will be beyond 60 by 2020 [4] and by 2050 it will even reach 37% [5]. Hence, the need of developing an autonomous, intelligent home care system will become more urgent with the steadily growing number of elderly people.

In an AmI environment, different sensors are installed to gather personal information. After data analysis in the AmI server, the status of the person can be estimated and the AmI system can provide appropriate help and predict emergencies, which then may be avoided by means of warnings. With the development of technologies, the importance of human-machine interaction in an AmI system increases steadily [6, 7]. It has been shown that robotic assistance at home is one potential way to alleviate the increasing pressure in health care systems [8].

In the KSERA project we are developing a socially assistive robot that supports some activities of daily life as well as health care needs of an elderly person, especially 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 [9] 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 the absence of a care giver, the Nao robot can assist a person's life by bringing medication, providing useful information, displaying videos using a portable beamer or supporting a video communication. To achieve this, the robot must first navigate to the person's position and a robust person tracking system is therefore important.

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 [10, 11] or radio waves require the person to carry certain technical devices, which may not be done reliably in everyday situations. Motion sensors [12, 13] 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 with these approaches, a vision system promises to provide good performance and 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 fall [14]. Hence, for tracking a person, a color camera is our main sensor of choice. Privacy concerns of camera surveillance are addressed in that no image information will be stored, since only the person's location is needed to be stored 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 different shapes at different locations, thus it is difficult to be recognized by static patterns (Figure 1). 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 the 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. We consider that different visual information sources can be used in combination to detect and localize a person's position reliably. A single ceiling-mounted camera with a fish-eye lens is used to keep the system simple and easy to install. A hybrid knowledge-based architecture integrates the different visual streams into a Sigma-Pi network architecture [15]. The system is able to start localizing a person with some of the cues and acquires the other cues online for the localization. The reliabilities of cues, which indicate the importance of this cue for decision making, can also be adapted. A particle filter [16] updates the person's position based on the output of this network.

Figure 1: Person images from a ceiling-mounted camera. It is hard to define a person by a fixed shape pattern

When the person is localized, the Nao robot will then assist by navigating to and interacting with the person. Consider that a service robot that shares with humans the same space needs to be socially acceptable and effective during the interaction [17], the physical embodiment of robots should make it likely that they will have to exhibit appropriate non verbal interactive behaviours [17], such as approach, touch, and avoidance behaviors. A behaviour-based architecture for the Nao robot is therefore presented, in which behaviours also encode non verbal interactions. Our behaviour-based model allows the robot to safely navigate in a cluttered and dynamically changing domestic environment, encodes embodied non-verbal interactions: the robot respects the user's personal space (PS) by choosing the appropriate distance and direction of approach. User preferences are described in a mathematical model that is input for the behaviors. The model is robust against unmodeled instances of the domestic environment.

The context and related works of the localization and navigation are introduced in section 2. The detailed system description is presented in sections 3 and 4. The experimental results are shown in section 5 with an evaluation and section 6 concludes this deliverable report.

2 The context and related work

2.1 Person localization based on ceiling-mounted camera

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 [18]. The precision of localization based on such status information is very low, while on the other hand, laser and stereo vision [19] offer high precision, however, at a high cost. Other suggested additional devices are motion sensors worn by the tracked person [20], using correlation of the motion sensor's signal with the motion registered by the camera. Person tracking based on multiple sensors [21, 22] 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 [23, 24] 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 [25] to

compensate for the narrow field-of-view of a single camera [26], or to overcome shadowing and occlusion problems [27]. Although these systems can detect and track multiple persons, these multi-camera systems 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. [28] 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. Moreover, with a wide-angle view lens, for example a fish-eye lens, the ceiling-mounted camera can observe the entire room. Occlusion is not a problem since the camera is at a position to see a person at any position within the room. But 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 [29], which finds the person based on the difference between an input and a reference image. Appearance-based models have been researched in recent years. Principal component analysis (PCA) [30] and independent component analysis (ICA) [31], for instance, represent the original data in a low dimensional space by keeping major information. Other methods like scale-invariant feature transformation (SIFT) [32] or a speeded up robust feature (SURF) [33] detect interest points (for example using Harris corner [34]) 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 [35, 36, 37] can detect a person precisely, but requires a very clear body shape captured from a front view. A multiple camera system has to be installed in a room environment to keep obtaining the body shape. The color obtained from the clothes and skin can be a reliable tracking feature [38, 23, 39], but this may have to be adapted quickly after changes.

In KSERA 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 [40]. 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 acts as a memory and enables the system to select a specific person for tracking. A particle filter approach [41, 26, 42, 43], which has potential for tracking, is developed to localize the person based on visual cues, which are being adaptively combined in a Sigma-Pi network architecture.

2.2 Robot navigation

Since the last decade control algorithms for robot navigation have started to include non-verbal interaction conventions, and several HRI trials have been conducted to derive quantitative models of them. Althaus et al. developed a behavior based architecture that allows a robot to join a group of people engaged in a conversation [44]. Nakauchi and Simmons developed a control algorithm that allows a robot to stand in line using a model of personal space (PS) derived from observation of people standing in line [45]. Yamaoka et al. presented a work on the definition of a model of the O-Space for a scenario where a robot is presenting an object to a person [46]. Walters et al. describe an empirical framework for determining the proxemic distance between a person and a robot, deriving the results from HRI trials [17]. Dautenhahn et al. describe user preferences in terms of approaching distance and direction for a robot offering a drink to a seated person [47]. Oskoei, Walters and Dautenhahn developed an autonomous robot that is able to adjust the proxemic distance between itself and a person [48].

A novel behavior-based architecture for the Nao robot will be proposed in section 4. Non-verbal interactions are also encoded into the behaviour. We formulate a mathematical model which allows to easily express user's preferences in terms of approaching distance and direction. The model is input for the definition of the robot final destination. We then provide a tool for adapting the model to the robot perception of the user's proximity.

3 Hybrid probabilistic neural model for person tracking

3.1 Tracking system overview

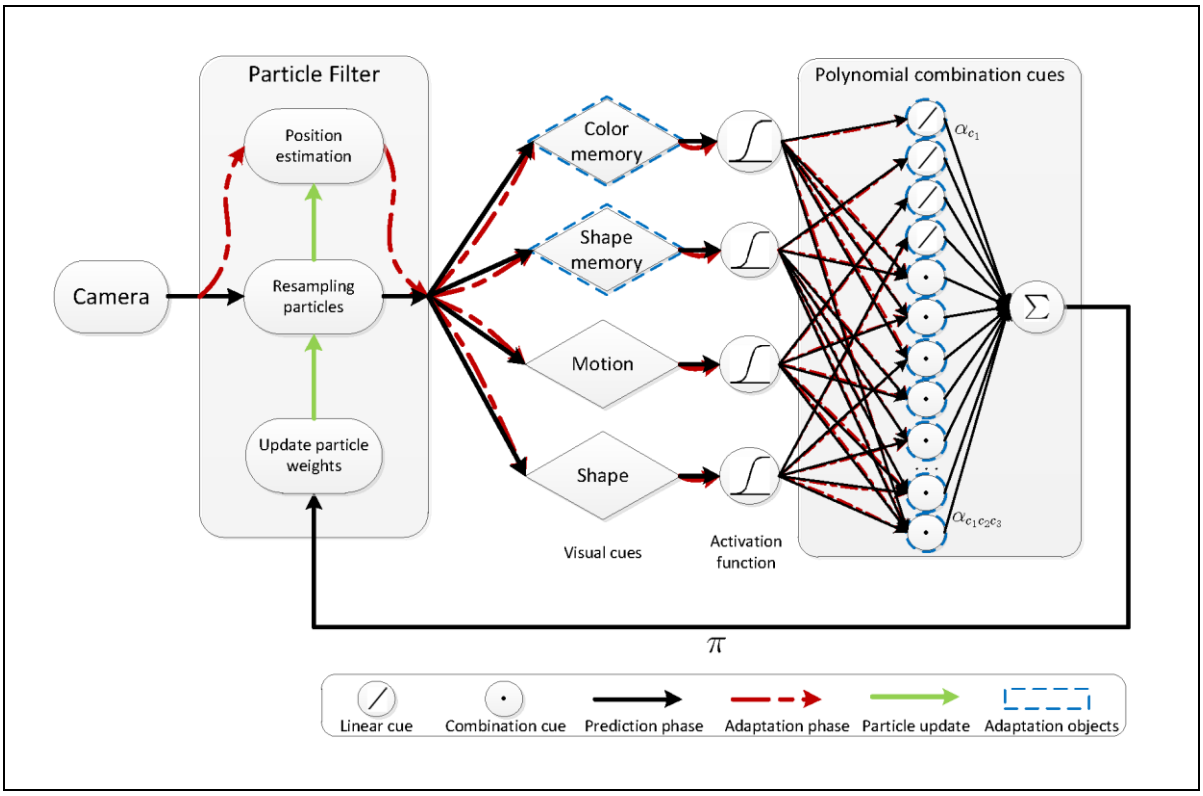

Figure 2: System overview

In the KSERA home scenario, a person and a small humanoid robot are being tracked. Our tracking model is illustrated in Figure 2. A Sigma-Pi network architecture integrates shape, motion, color memory and shape memory streams and feeds its output to a particle filter, which provides robust object tracking based on the history of previous observations [49, 16]. The work flow can be split into two parts: *prediction* and *adaptation*.

In the prediction phase (black arrows in Figure 2), 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. Here we use the sigmoid function as the activation function, which can be described with Eq.(1):

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

where x is the function input and g is a constant scale factor. Through the polynomial combination of cues represented by a Sigma-Pi network, the weights of particles are computed. The particle will then be resampled and the position of the particles will be updated (green arrows in Figure 2). 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 Figure 2). The color memory cue and shape memory cue will be learned and the reliabilities of visual cues, which will be described in section 3.4, will be adapted based on the validation results (labelled with the blue dashed lines). With the collaborative contribution of each cue, the tracking performance can be improved significantly.

3.2 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) [50]. 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 person's actions. Based on the Bayesian representation in a POMDP, the agent's state can be estimated according to an iterative equation:

$$P(s_t|z_{0:t}) = \eta P(z_t|s_t) \int P(s_{t-1}|z_{0:t-1}) P(s_t|s_{t-1}) \mathrm{d}s_{t-1} \tag{2}$$

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 how likely the state is given the existing observations, 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 $P(s_{t-1}|z_{0:t-1})$:

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

In the particle filter model, 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 following form:

$$P(s_t|z_{0:t}) \approx \sum_i \pi_{t-1}^{(i)} \delta(s_t - s_{t-1}^{(i)}) \tag{4}$$

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; in our model we use the sequential importance resampling algorithm, which is described in the following Algorithm 1.

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:

$$\begin{aligned} s_t^{(j)} &\propto \pi_t^{(j)} \\ \pi_t^{(j)} &= \frac{1}{N}, \qquad \text{for} \quad j = 1 \ldots N \end{aligned}$$

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:

$$P(s_t|s_{t-1}^{(i)}, a_{t-1}) = \frac{1}{\sqrt{2\pi\sigma(a)^2}} e^{-\frac{(s_{t-1}^{(i)} - s_t^{(i)})^2}{2\sigma(a)^2}} \tag{5}$$

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 3.4.2) in the action variable a_t, however, is nevertheless 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:

$$\sigma(a) = \begin{cases} v_1 & \text{if motion detected} \\ v_2 & \text{else} \end{cases} \tag{6}$$

where v_1, v_2 are constant parameters with $v_1 > v_2$. When no motion is detected, the probabilistic distribution will then shrink to a small area that allows the particles only to move close to the previous position. This models the behavior 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 pixel-wise search window methods.

3.3 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 [15]. 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:

$$\begin{aligned}\pi^{(i)} = &\sum_{c}^{4} \alpha_c^l(t) A_c(s_{t-1}^{(i)}) + \sum_{c_1>c_2}^{4} \alpha_{c_1c_2}^q(t) A_{c_1}(s_{t-1}^{(i)}) \\ &A_{c_2}(s_{t-1}^{(i)}) + \sum_{c_1>c_2>c_3}^{4} \alpha_{c_1c_2c_3}^c(t) A_{c_1}(s_{t-1}^{(i)}) \\ &A_{c_2}(s_{t-1}^{(i)}) A_{c_3}(s_{t-1}^{(i)})\end{aligned} \tag{7}$$

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 [51]. The coefficient 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. 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, is 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 [52, 40]. An issue is that an adaptive cue will be initially unreliable, but when adapted 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 cue's reliability α. The details of each visual cue will be introduced in the next section.

3.4 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 [53]. 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 backprojec-

tion algorithm [54]. The *shape memory cue* is based on a set of SURF features [33] weighted by the correlation with adjacent frames. The details of these methods are introduced in the following paragraphs.

3.4.1 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. In the shape cue, the image patch of the particles is pre-processed by a Laplace filter with 3×3 pixel kernel and converted to a counter image (see Figure 3). 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 at first. The data collection contained 6000 images. 75 percent of the data are used for learning and 25 percent for the testing. After the training phase, the network is able to classify new images. For each particle, we take the winner of the neural network output and learn the shape cues if the winner matches the group of the particles. This process is shown in Figure 3.

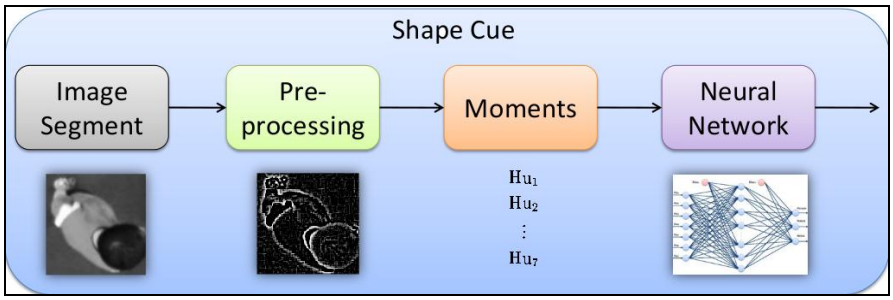

Figure 3: Process of the shape cue

In computer vision, moment invariants are essential to analyze or recognize objects independent of their position, rotation or scale [53, 55]. Because the person's shape, i.e. the counter image 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-moment [56] provides therefore a good method to solve this problem. For a $M \times M$ grey-value image, the two-dimensional moments can be given as:

$$M_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} x^p y^q f(x,y), \qquad p,q = 0,1,2,\ldots \tag{10}$$

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 centre of the mass $(\bar{x},\bar{y})$ with:

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

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

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

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

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

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

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

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

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

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

To detect the person from the moment invariants, we train a multilayer perceptron [57, 58] network for the classification. This artificial neural network consists of multiple layers of neurons, which are connected fully with the neurons in the neighbour layers. An MLP network can be used for function approximation and classification based on supervised learning. The MLP for shape classification is shown in Figure 4. Seven input neurons connect directly with the Hu moments. In the middle layer we use 30 neurons with the sigmoid activation function.

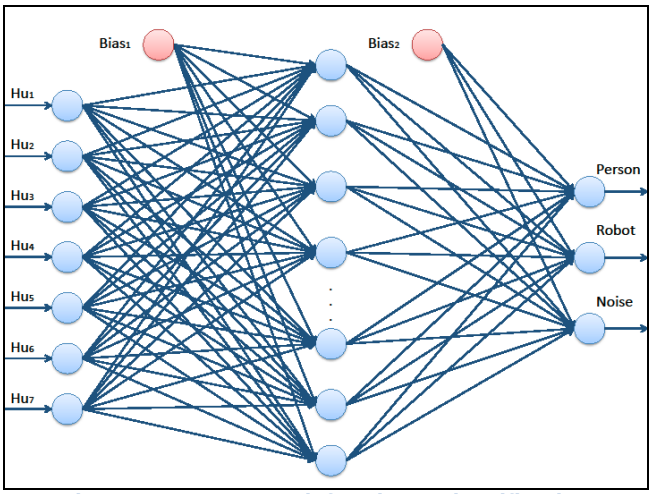

Figure 4: MLP network for shape classification

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 winner of the output neuron $\text{win} = \arg_i \max(y_i^{\text{out}})$ returns the classification result. If it is the same as the group index of particle filters, for instance the output of the MLP network shows the shape of a particle is "person" and this particle belongs also to the group "person", the shape cue will receive a positive feedback. Then the reliability $\alpha_s(t)$ of the shape cue will be updated according to Eq.(8).

3.4.2 Motion cue

Motion detection is a method to detect an object by measuring difference in the image. We use here the background subtraction method [29] 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 first 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:

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

The pixels are merged with blob detection which allows that the connected pixels are labelled with the same blob index. The motion objects are segmented with this method. Considering that the background may also change, as when moving the furniture, the background is updated smoothly with the following formula:

$$i'(\mathbf{x}, t) = (1-\gamma) i'(\mathbf{x}, t-1) + \gamma i(\mathbf{x}, t) \tag{21}$$

where $\gamma \ll 1$ is an update rate. When the new input image remains static for a long 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 he will not be detected anymore. In this case, the other visual cues will allow the system to find the person.

3.4.3 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 people does not change quickly, it is a reliable feature for tracking.

A large number of tracking methods are based on color information [59, 60, 39]. 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 colours are converted to the HSV space [61]. 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:

$$R_i = \min\left(\frac{O_i}{I_i}, 1\right) \tag{22}$$

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:

$$O_i(t) = (1-\zeta)O_i(t-1) + \zeta I_i(t) \tag{23}$$

where ζ here denotes an update rate. The ratio histogram R_i represents then 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 colour index. 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.

Considering that the tracked person might wear clothes with different colours 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 colours of a person do 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.

3.4.4 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 here SURF features [33] 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 neighbouring 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 structure of this design is shown in Figure 5.

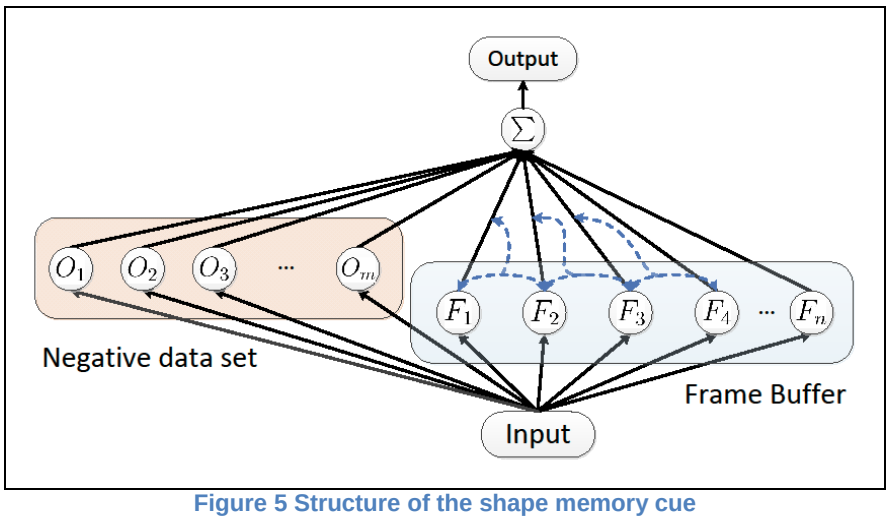

Figure 5 Structure of the shape memory cue

4 Robot navigation based on user's proximity for Bayesian inference

Navigation is a necessary capability for mobile robots to move around in their surroundings. This issue is a widely studied research topic and a large number of strategies and systems have been developed for all kinds of environments. Most of these studies focus on reaching a certain goal point in a given environment while avoiding collisions on the way [44].
When the robot is supposed to live in a domestic environment, sharing the same space with a human being, its navigation trajectories represent a non-verbal communication cue that influences the quality of interaction. While map-based approaches define the robot trajectories as a function of the map, behavior-based techniques define the robot overall behavior independently of the map. In a cluttered home environment that has to be dealt with in KSERA, therefore, a behavior-based technique is preferred.
Each individual behavior solves a navigational subtask, for example avoiding obstacles, reaching the goal point or adapting the orientation to face the human [44]. Subtasks' coordination results in the overall robot behavior [62]. Thus the state of the robot is not determined by the property of the world map directly but it depends on how the behaviors are defined.

4.1 Behavior definition

The behaviors' definition follows from the work by Bicho [63], Soares et al. [64], and Althaus and Christensen [44] and its formal background is based on the theory of non-linear dynamical systems [65]. A behavior emerges from the time evolution of a behavioral variable $\varphi(t)$ that is chosen to be the robot heading direction. Thus a behavior is represented by a non-linear dynamical equation:

$$\dot{\varphi}(t) = f(\varphi(t)) \tag{24}$$

where $f(\varphi(t))$ can be interpreted as a force acting on the behavioral variable $\varphi(t)$ [44]. Multiple behaviors are aggregated by means of a weighted sum:

$$\dot{\varphi}(t) = \sum_{t=1}^{m} w_i f_i\left(\varphi(t)\right) + \text{noise}. \tag{25}$$

Behavior competition/coordination is realized by shaping the values of the weights w_i. The architecture presented here defines four elementary behaviors.

4.1.1 Avoid obstacles

This behavior moves the robot away form the obstacles generating a repulsive force that is exponentially dependent on the detected distance between the robot and the obstacle and on the angular location of the obstacle with repsect to the robot. Its mathematical expression is given by:

$$f_{\text{obs}} = \beta_1 \exp\left(-\frac{1}{\beta_2}(d_{\text{obs}-R})\right)(\varphi - \psi)\exp\left(-\frac{(\varphi - \psi_{\text{obs}})}{2\sigma_{\text{obs}}}\right) \tag{26}$$

The terms β_1, β_2, σ_{obs} are coefficients that determine the repulsion strength of Eq.(26). High values result in a high value of f_{obs}. The detected distance between the robot and the obstacle is represented by the term d_{obs} while the direction of the obstacle respect to the robot is encoded in the term $\varphi - \psi_{\text{obs}}$. The term R represents the robot radius.
The behavior is in competition with the behavior that makes the robot respect the user's intimate space. A conflict arises when an obstacle and the user's intimate space overlap. The weighting

function accounts for their competition regulating the activation of the behavior. Its expression is given by:

$$w_{\text{obs}} = \begin{cases} 1 & \text{if } d_{\text{obs}} < d_{\text{ps}} \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

where the distance between the robot and the user's intimate space is given by d_{ps} .

4.1.2 Reach Target

The robot must move towards a point in the personal space to be able to face the human. This is modeled by an attractor dynamic with expression:

$$f_{\text{tar}}(\varphi) = -\lambda_{\text{tar}} \sin(\varphi - \psi_{\text{tar}}) \quad (28)$$

The parameter λ_{tar} represents the strength of the target attractor dynamic while the term $\varphi - \psi_{tar}$ accounts for the angular location of the target with respect to the robot. This behavior is always active, hence the value of its weighting function is always one.

4.1.3 Avoid close intimate zone

Personal space (PS), or more broadly proxemics, is the "bubble" of space that people attempt to keep around themselves and others [66, 67]. This behavior is modelled by a circular repulsive force around the user that makes the robot avoid his close intimate zone. The parameters of the user's close intimate zone are taken form Lambert [68] and are reported in Table 1.

Table 1: Human-Human Personal Space Zones [68]

Range	Situation	PS zone
0-15 cm	Lover or close friend touching	Intimate Zone
15-45 cm	Lover or close friends	Close Intimate Zone
45-120 cm	Conversation Between friends	Personal Zone
120-360 cm	Conversation to non-friends	Social Zone
more than 360 cm	Public Speech making	Public Zone

The repulsion strength is locally attenuated by the weighting function to permit the robot to closely approach the user when it is requested to. Its expression is given by:

$$f_{\text{ps}} = \beta_3 \exp\left(-\frac{1}{\beta_4}(d_{\text{ps}-R})\right)(\varphi - \psi_{\text{ps}}) \exp\left(-\frac{(\varphi - \psi_{ps})}{2\sigma_{\text{ps}}}\right) \quad (29)$$

The construction of expression (29) is very similar to Eq. (26) as are the parameters. The term $\varphi - \psi_{\text{ps}}$ determines the direction from which the distance between the robot and the personal space d_{ps} is evaluated.

This behavior is in competition with the behavior "avoid obstacles" when an obstacle coincides with the personal space. The conflict is again modeled by the shape of the weighting function w_{ps} , which also models the field attenuation in the correspondance of the location of the target point. The weight expression is given by:

$$f_{\text{ps}} = \begin{cases} \left(1 - e^{-\frac{(\theta_r - \theta_t)^2}{\sigma_{sd}}}\right) & \text{if } d_{\text{ps}} < d_{\text{obs}} \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

where the terms θ_r represents the robot angular location in the user's reference frame and the term θ_t represents the target angular location in the user's reference frame. Their difference $(\theta_r - \theta_t)$ expresses how close the robot is to its target in terms of angular location.

4.1.4 Align to user

When the robot is close to its target point, which is the point from which the interaction with the user is supposed to start, it should adapt its trajectory to face him. This behavior is also modeled with an attractor dynamic:

$$f_{\text{au}}(\varphi) = -\lambda_{\text{au}} \sin\left((\varphi - \psi_{\text{user}}) - (\theta_{\text{tu}})\right) \tag{31}$$

where the term λ_{au} represents the strength of the alignment force. The term $\varphi - \psi_{\text{user}}$ represents the robot orientation with respect to the user and the term θ_{tu} represents the desired robot final orientation with respect to the user.
The alignment becomes particularly relevant when the robot has an antropomorphic shape because the front and the back are clearly recognisable. The actractor dynamics of the alignment behavior is undesirable when the robot is far from the target point. This consideration is modeled by the weighting function that decreases the attractor dynamic of the alignment behavior exponentially with the distance between the robot and the target point:

$$w_{\text{al}} = \exp(-d_t) \tag{32}$$

where d_t represents the distance between the robot and the target point.

4.1.5 Behavior evaluation

Given a target point the behaviors are able to safely move the robot in a cluttered environment. Figure 6 shows a MATLAB™ [69] simulation of the overall behavior. The robot model uses two sonar sensors to provide distance measurements. The sensor locations are based on those on the robot NAO [9] .

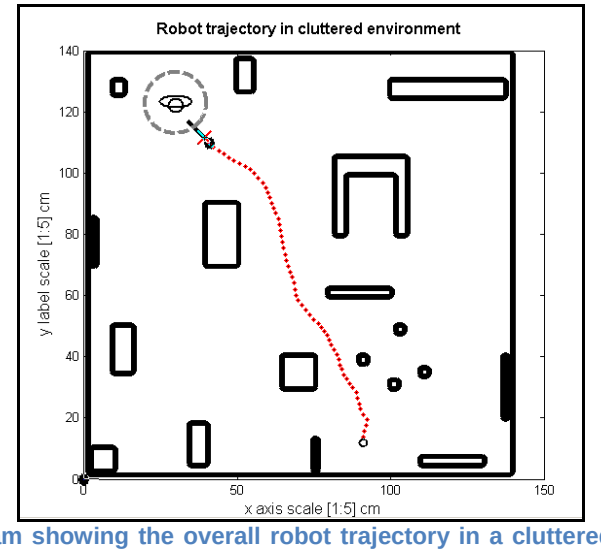

Figure 6 Diagram showing the overall robot trajectory in a cluttered environment (dots). The robot safely avoids the obstacles heading towards the target point (cross). When it is close to the target point it smoothly modifies its trajectory to safely face the user. The user is represented at the center of the dashed circle and

the dashed circle represents the close intimate space. The final robot orientation is highlighted by a bold line that points in the direction of the user.

4.2 Target representation

The robot is driven by an attractor dynamic that steers it towards the target point. We now focus our attention on the definition of the target point when the robot is requested to approach a person to convey information in a domestic environment.
Several studies have been conducted to assess how close people prefer the robot to approach them and from which direction, see [47], [17], [48] and [67]. The results of these studies have already been implemented in robotics control systems. Sisbot et al. formulated a constraint optimization problem to include findings on human-robot proxemics in their navigation framework [70]. But their approach is not suitable for a behavior based control architecture because the robot trajectories are derived as a property of the map trough the formulation of the constraint optimization.
Pacchierotti et al. [71] proposed an interaction model for a robot encountering a person in a corridor. Their model refers to the avoidance behavior of the robot in a casual encounter between it and a person. Their model is not suitable for the problem addressed by this work for two reasons. First we are primarily interested in the approaching behavior of the robot with respect to a person and not in the avoidance behavior. Second, because their work refers to casual encounters, while we are interested in modelling the approaching behavior in a domestic environment, where the person and the robot are constantly sharing the same space.
Kirby et al. [67] formulated a model that allows a mobile robot to respect the user personal space and to tend to one side of a hallway. They work refers to the avoidance behavior of the robot when encountering a person in the hallway. The social constraints are expressed in terms of constraints in a constraint optimization problem that is not suitable for the behavior-based architecture defined in Section 4.1. It also addressed the avoidance behavior and not the approaching behavior in which we are interested.
Oskoei, Walters and Dautenhahn proposed an autonomous robot that controls its proxemic distance form a person. Their system is lacking the ability to adapt the model to the presence of obstacles in the user proximity and does not take into account the direction of approach but just the distance [48].
Here we formulate a model of the user personal zone (see Table 1) that allows encoding user preferences in terms of approaching distance and direction. The model is suitable for statistical computations in order to adapt it to the robot perception of the environment in the proximity of the user.

4.2.1 Approaching Model

The model aims at describing what the user expects from the robot in terms of approaching distance and direction independently from the properties of the environment in the user's proximity. The model is also independent from the algorithm used for solving the robot navigation problem. The proposed model also deals with robustness. It allows the robot to adapt its behavior in the presence of unmodeled instances of the real world. Indeed robustness should be one of the key issues in designing models of social constraints for mobile robots. When defining the approaching model two main design issues were taken into account. An easy translation of the results of HRI trials into the model. And robustness against possible unmodeled obstructions in the user proximity. Robustness can be achieved by the inclusion of tolerances in the model yielding to the representation of a region of space rather than a single point.

4.2.1.1 Model definition

In the literature, the direction and distance of approach are usually expressed with respect to the person [17], [47]. In analogy, we formulate the model with respect to a reference frame centred on

the user head with the y-axis defined as straight ahead and the x-axis rightwards, see Figure 7 and Figure 8.

The region of approach can be defined as a function of the approaching distance ρ and the approaching angle θ. Since we are interested in modelling a region of space we also introduce the tolerances on the distance, namely σ_ρ , and on the angle, namely σ_θ . Given a desired approaching distance ρ_0, a desired approaching angle θ_0, and their tolerances σ_ρ and σ_θ, the model can be defined as:

$$\Phi(\rho,\theta) \sim \exp\left(-\frac{(\rho-\rho_0)^2}{\sigma_{\rho_0}^2}\right)\exp\left(-\frac{(\theta-\theta_0)^2}{\sigma_{\theta_0}^2}\right) \tag{33}$$

A representation of the model for an approaching angle of 45 deg with tolerance 5 deg, and an approaching distance of 75 cm with tolerance of 40 cm on the distance can be seen in Figure 7.

Table 2 Parameters values for the probability distribution shown in Figure 8.

Parameters settings					
k	λ_k	ρ_k	$\sigma_{\rho_k}^2$	θ_k	$\sigma_{\theta_k}^2$
1	$\frac{1}{2}$	70 cm	40 cm^2	45 deg	5 deg^2
2	$\frac{1}{5}$	70 cm	40 cm^2	90 deg	5 deg^2
3	$\frac{3}{10}$	70 cm	40 cm^2	135 deg	5 deg^2

The target location is defined as the point (ρ,θ), expressed with respect to the user reference frame that leads to the maximum of the distribution (33).

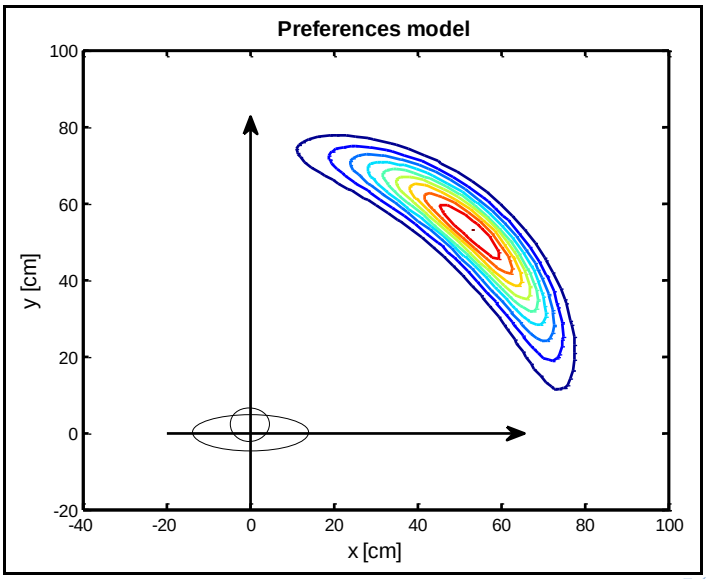

Figure 7: Contour graph showing the user preference model $\Phi(\rho,\theta)$ in Eq.(33) with parameters $\theta_0 = 45\text{deg}$, $\sigma_{\theta_0}^2 = 5$ deg , $\rho_0 = 75$ cm and $\sigma_{\rho_0}^2 = 40\ \text{cm}^2$. The function $\Phi(\rho,\theta)$ presents a peak at (ρ_0,θ_0) and decreasing values around the peak.

4.2.1.2 Expressing multiple preferences

The model introduced in Eq. (33) can easily represent a single preference with related tolerances. But it is possible to further develop the model representing not just one preference but multiple ones. Given a set of preferences configurations $\{(\rho_0, \sigma_{\rho_0}, \theta_0, \sigma_{\theta_0}) \dots (\rho_k, \sigma_{\rho_k}, \theta_k, \sigma_{\theta_k})\}$ multiple preferences are expressed introducing $\Phi(\rho, \theta)$ of Eq. (33) in a weighted sum:

$$F(\rho, \theta) = \sum_{k=1}^{n} \lambda_k \Phi_k(\rho, \theta) \tag{34}$$

where the index k represents the particular configuration to be modelled and λ_k is a measurement of the degree of the "desirability" associated to the configuration k. Results of HRI trials can be easily expressed by means of equation (34). As an example consider the study performed by Dautenhahn et al. [47] and Walters [17] regarding proxemic distances and approaching directions between a robot and a person. The results of their studies showed that a seated person prefers to be approached from one side with a slightly preference for the right [47]. These findings can be modelled using expression (34). Three configurations for approaching angles of 45, 90 and 135 deg were considered. The weights in Eq. (34) can be used to quantify the "desirability" of each of the three configurations. Practically it is possible to weigh the approach from 45 deg. slightly more than the approach from 135 deg while weighting them both more than the frontal approach at 90 deg. Figure 8 shows the model evaluation in the proximity space of the user. The parameters for the three configurations are reported in Table 2.

The model is quite appealing because it allows expressing findings from HRI trials in an easy way in a form that can be used in statistical computations. The key element is the representation of tolerances that can be used to increase the robustness of the overall navigation architecture. Indeed each point in the space encodes a "desirability" value according to the acquired knowledge on user preferences and their tolerances. A high value indicates a high desirability of the location and *vice versa*.

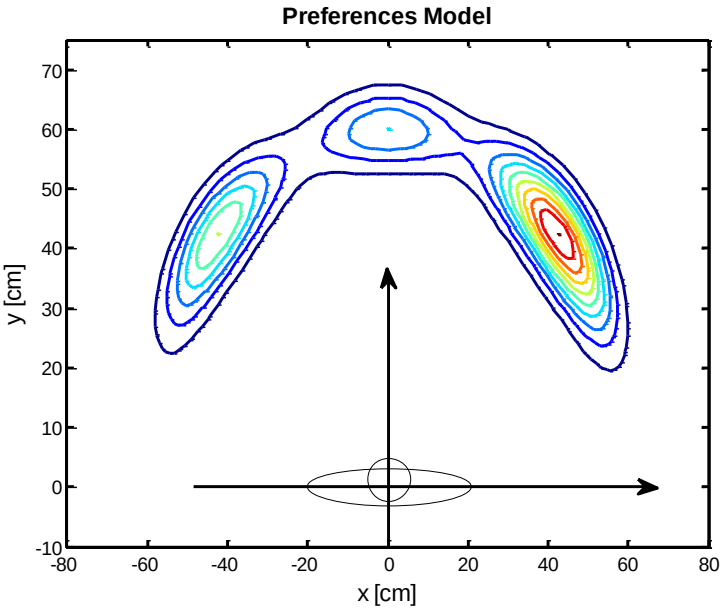

Figure 8: Contour graph showing the preferences model described in Eq. (34) for three approaching sets. The model parameters are described in Table 2. The contour plot shows that the right side approach has higher values of the function and thus it expresses a stronger "desirability". And that between a frontal approach and a left side approach, the latter is more desirable.

4.3 Location inference

.

The model introduced in expression (34) is suitable for Bayesian inference with the objective of deriving the best robot position in the proximity of the user given the knowledge about user preferences and the robot perception of the environment.

State estimation using Bayesian filters such as Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Particle Filter (PF) has been widely used in robotics primarily for SLAM or localization algorithms (see [72] and [73] for an overview). The state to be estimated is usually the robot pose in the environment or both the robot and the human pose respect to it.

While the KF, EKF and UKF propagate Gaussian distributions representative of the posterior, the PF propagates multimodal distributions [73]. The latter property of the PF is appealing when formulating the problem of inferring the best location of the robot in the user's proximity space, because this can be initially represented by a multimodal distribution as already described in Section 4.2 and represented in Figure 8. Therefore we formulate the problem of estimating the direction and distance of approach as a Bayesian filtering problem and we solve it by mean of a particle filter.

4.4 Problem formulation

The general Bayesian filtering problem, at time t, consists of computing the posterior distribution $P(s_t|z_{0:t-1})$ of the hidden state (s_t) of a dynamical system given observations (z_t) and control inputs (u_t) [72],[62]. The posterior distribution, or belief, of the state $P(s_t|z_{0:t})$ is computed according to Eq.(2). Respect to the problem that we are addressing, the dynamical system is composed by the user, the robot and the obstacles in the world. The system state s_t, or in other words the quantity to be inferred from the observations by means of the filter, is the most "desirable" robot location with respect to the user in terms of approaching distance and angle. Thus the state is given by:

$$s_t = \begin{pmatrix} \rho_t \cos(\theta_t) \\ \rho_t \sin(\theta_t) \end{pmatrix} \tag{35}$$

where the terms ρ_t and θ_t represent the coordinates of the point with maximum "desirability" expressed in the user reference frame. The term y_t represents the measurements available at time t and it accounts for the perceptual cues that the robot has about the environment around the user. The term u_t accounts for the user movement in the space. The particle filter algorithm is applied to the recursive Bayesian filtering problem and it is reported in pseudo-code in Section 3.2.

The initialization phase of the algorithm generates a uniform distribution of N particles positioned in the proximity of the user with the following distribution:

$$s^{(i)} = \begin{pmatrix} \rho_i \cos\theta_i \\ \rho_i \sin\theta_i \end{pmatrix} \tag{36}$$

where ρ_i and θ_i are distances and angles in the user reference frame sampled from uniform distributions. The particles are then referred to a global reference frame through a coordinate transformation. The weights w^i are assigned to each particle according to the evaluation of expression (33). During the iteration phase, the algorithm first evaluates the motion model $p(s_t|s_{t-1}, u_{t-1})$ that accounts for the user movements in the space. The motion model generates s_t^i from s_{t-1}^i shifting the particles in the space according to the measured user movements.

The innovation term, $(z_t|s_t)$ expresses the likelihood that a particle represents a "desirable" approaching distance and direction according to the robot perception at time t. At a first glance, one might say that a particle is desirable when it leads to high value of the preferences model in Eq. (34). Unfortunately this is not true in at least two situations. The first case is represented in Figure 9. The robot is approaching the user from the left side, according to the preference model it should turn left, cross the region in front of the user and arrive at his right side while aligning to face him.

This trajectory would not show much robot intelligence and would be uncomfortable for the user. The second case is represented in Figure 10. The robot is approaching the user from the right side looking for the maximum of the preferences distribution. On the user's right side is also present an obstacle which makes it impossible for the robot to reach its goal. These two examples show the necessity to include the robot's perception of the environment in the measurement equation of the particle filter to allow the robot to cope with unforeseeable instances of the reality in the user's proximity. From the latter consideration the expression for the measurement equation of the particle filter is derived:

$$w^i = \frac{1}{d(s_t^i, s_r)} F\left(s_t^i\right) = cc_1 \cdot cc_2 \tag{37}$$

where $d(s_t^i, s_r)$ represents the distance between the robot and the particle s_t^i and $F(s_t^i)$ represents expression (34) evaluated at the particle location expressed in polar coordinates with respect to the user's reference frame. It is straightforward to notice that the introduction of the term $d(s_t^i, s_r)$ in the weighting expression (37) allows coping with both situations represented in Figure 9 and Figure 10. The conceptual novelty of the particle filter presented here is the fusion of contextual cues derived from the robot's perception with knowledge derived from psychological experiments. Expression (37) accounts as contextual cue the particle's proximity to the robot but the weight expression could take into account multiple contextual cues i.e. the user's head pose. The multiplication of the terms in Eq.(37) indicates that a particle is positioned in a desirable location if it simultaneously represents a feasible point (thus an high value of cc_1) and a desirable point in terms of preferences (thus an high value of cc_2).

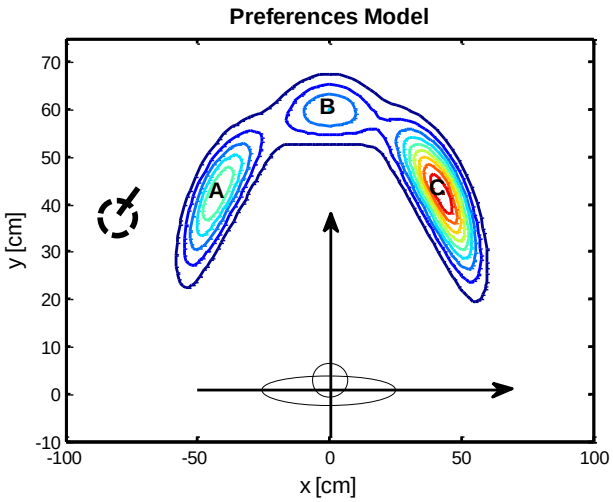

Figure 9: Diagram showing the contour plot of the preferences model with the parameters retrieved from Table 2. The robot (dashed circle) wants to approach the person located at the centre. According to the preferences model the optimal target point is in region C because the maximum of the distribution is located there. However, going to point C causes the robot to cross the user's personal space which could produce an uncomfortable trajectory.

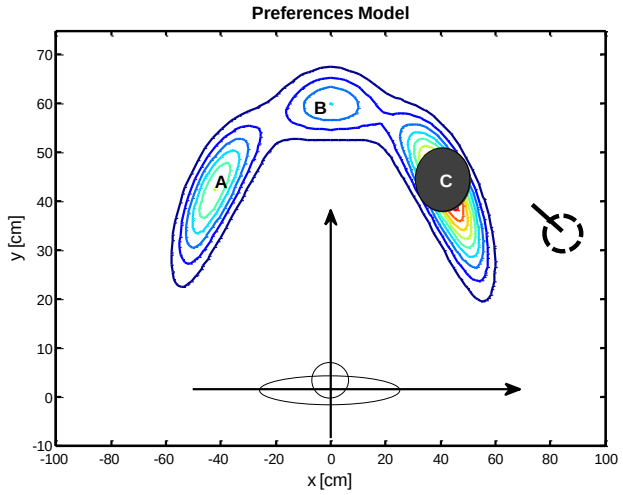

Figure 10: The robot is approaching the user from the right side where the maximum of the distribution is located. An obstacle is positioned exactly at the distribution peak. The robot cannot reach its target point, therefore it has to be relocated.

When the robot is facing the situation described in Figure 9, it evaluates the left side particles more than the others because they are closer to it and they present a high value of the preferences model in Eq.(34). The weighting expression (34) also allows dealing with the situation described in Figure 10. Indeed when the robot tries to position itself in the point with the maximum probability it is driven away by the behavior avoid obstacle in Eq.(26). Thus during its motion it gets close to other particles determining a shift of the local maximum of the probability distribution.

5 Experimental results and evaluation

In this section we will show our initial experimental results of the hybrid probabilistic neural system for person tracking and the robot navigation model using Bayesian inference. The KSERA lab as an experimental environment as well as test scenarios will be shown at first. The results will be indicated and evaluated then.

5.1 Hybrid probabilistic neural model for person tracking

The environment for testing the tracking system is shown in Figure 11. The individual frames show the KSERA lab from the ceiling camera. Its fish-eye lens is calibrated and the camera image is sub-sampled to the resolution 320×240, which allows real-time processing. 10 different videos have been tested. The experiment aims at detecting and locating a person or a mobile robot under static conditions in the image as well as at tracking their motion 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 trained color pattern, the shape and motion cue 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 [74]

Most of these tests have been carried out in our experiment room. 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 detailed descriptions of scenarios as well as their results are shown in the coming sections.

5.1.1 Experimental results

5.1.1.1 Tracking a person moving in the KSERA lab

A test example is shown in Figure 11. 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 lab (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.

5.1.1.2 Tracking a sitting person

A person can stay in a position for a long time for example while watching television, reading book, etc. In this case no motion will be detected thus the motion cue will temporarily not work. In this situation 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 sit down in our experiment and the particles can keep localizing the person for long time.

5.1.1.3 Changing light condition

In a real life home, 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 Figure 12. 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 are recovering and return to the position of the person (frame 115).

5.1.1.4 Changing furniture position

Another challenge is to modify the room structure inside the KSERA lab. The disturbance of a changing environment, for example a moving table in the room (Figure 13) 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.

5.1.1.5 Distracter person

The target of presented tracking system is to localize a single person, but it is common that several 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 Figure 14 we show a test scenario in which two persons walk across each other. A person is tracked at the test beginning (frame 317). When two people walk very close (frame 324), the particles are still able to keep tracking the target person. Figure 15 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 come back to the target person again (frame 423).

5.1.1.6 Distracter person using CLEAR 07 data set

A set of experiments based on the fish-eye camera video of CLEAR 07 short sample data set have been done to evaluate our tracking performance based on external data. The idea of our system is to monitor a target person when he is alone in the room. Because the CLEAR 07 multiple person tracking data set aims to track multiple person, our currently system will rely on selecting a target person. Therefore, we can only evaluate the system after when one of the person is tracked. The experiment results is shown in Figure 16. When a person is tracked successfully, the person will be localized always until the end of this video.

5.1.2 Evaluation

The experiment results have been evaluated principally according to the CLEAR MOT Metrics [74]. Since only a single person is tracked in the system, based on our goal design, the miss frames m and the false positives frames fp have been counted and the multiple object tracking accuracy (MOTA) have been calculated. The threshold distance of the false positive was defined as 40 pixels. 10 videos were evaluated and the results are summarized in Table 3. 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 Figure 12). 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.

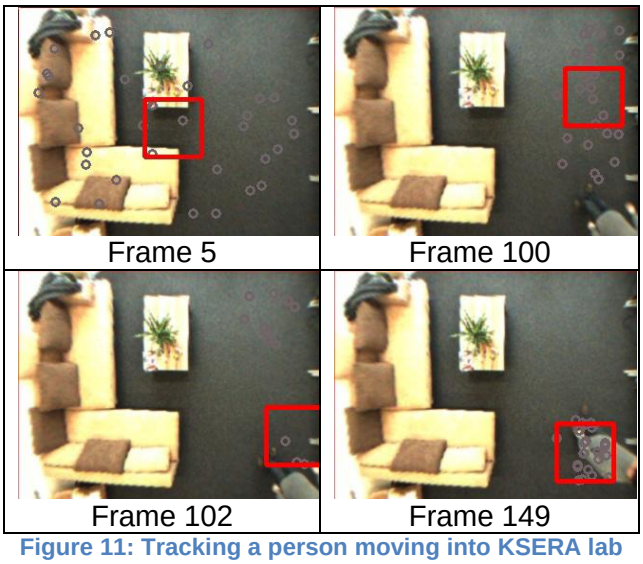

Figure 11: Tracking a person moving into KSERA lab

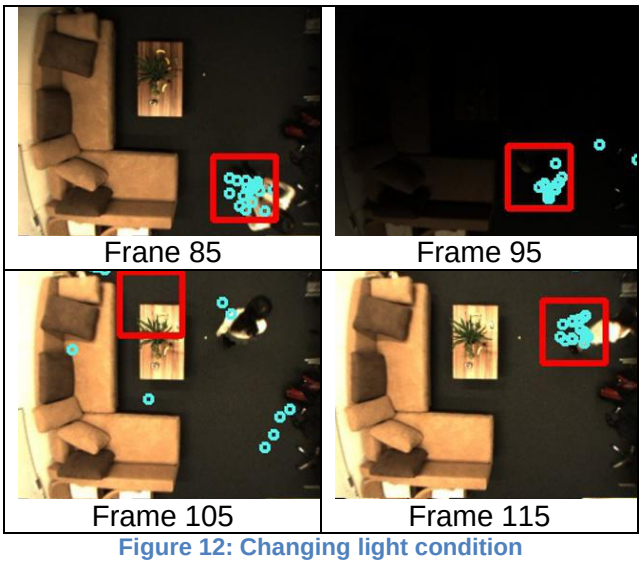

Figure 12: Changing light condition

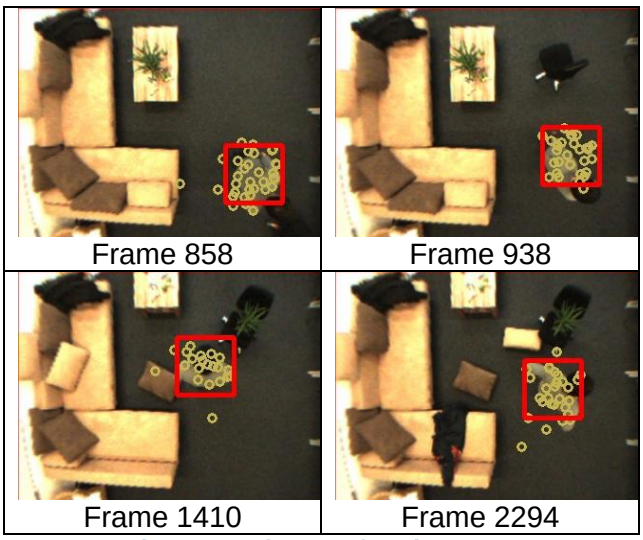

Figure 13: Change of environment

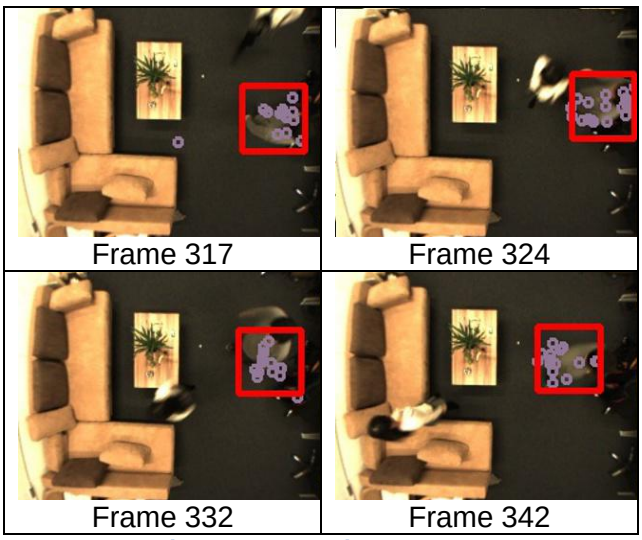

Figure 14: Crossing a person

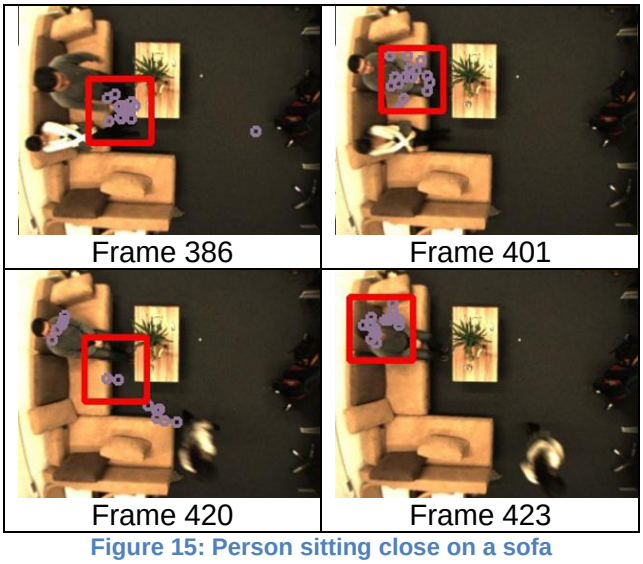

Figure 15: Person sitting close on a sofa

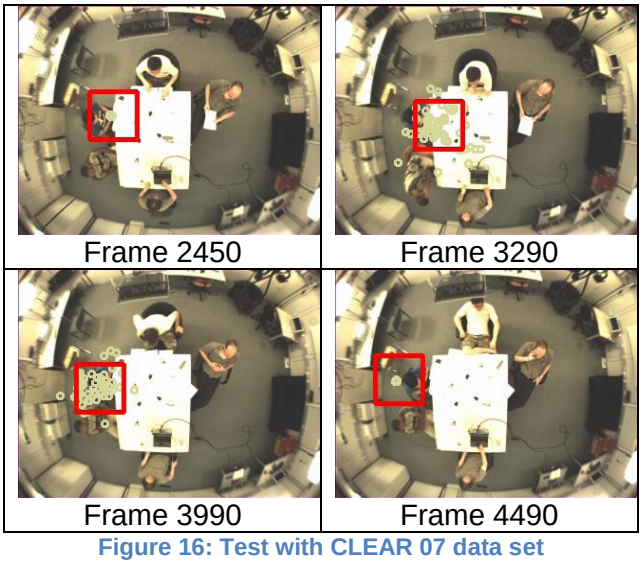

Figure 16: Test with CLEAR 07 data set

Table 3: 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**

5.1.2.1 Reliabilities

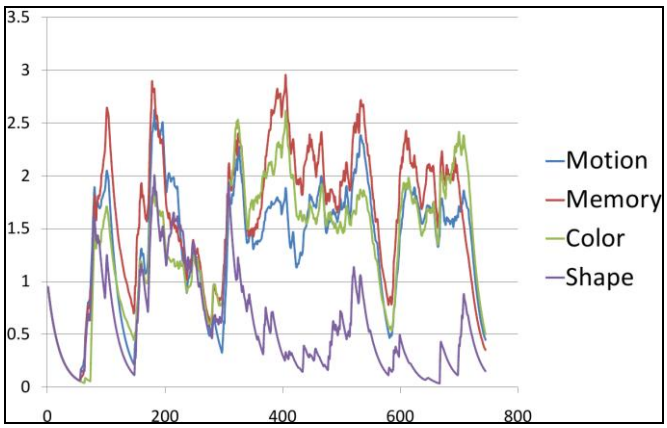

Figure 17: Reliabilities of linear cues

The contribution of visual cues can be evaluated by their reliability values. The more often a visual cue helps to find the person, the higher the reliability of this cue will get. The reliabilities of linear visual cues are shown in Figure 17. The x-axis of the diagram denotes the step 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.

5.1.2.2 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 pixel-wise search window method, because only a few particles (30) process a small part of the images (60×60 pixels) at each step. In Table 4 we list the computation time for 100 steps with different number of particles. The particles are able to track the person correctly throughout all these tests. Therefore this system is convinced to work under real-time condition.

Table 4: Computational time with different particle numbers

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

5.2 Robot navigation using Bayesian inference

The use of the particle filter to introduce the preferences model in the choice of the robot final location has been tested on simulation. The simulation is carried out in a MATLAB™ [69] simulation environment. User pose and robot pose are provided externally by the environment. The robot scans its environment by means of two sonar sensors. The sonar sensor locations on the simulated robot are modeled after the sonar sensor locations on the robot NAO [9]. The same applies to the sonar sensors' effective cone. The initial target distribution is given by three configurations whose parameters are expressed in Table 2 and it is visible in Figure 18.

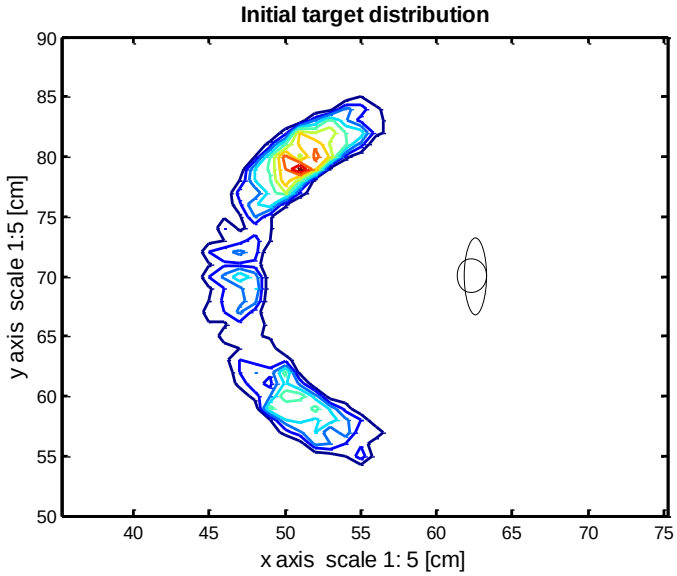

Figure 18: The figure shows the initial target distribution according to the preferences model in Eq. (34) with the parameter settings as expressed in Table 2. The user's location is indicated in the Figure at coordinates [62 70]. The particle filter gives an approximation on the real initial belief. Consequently, the shape of the distribution is different from the one in Figure 8. The number of particles used for expressing the initial belief is 2220. The initial target location is at the right side of the user at an angle of 45 deg as it is highlighted by the peak of the distribution at coordinate (51, 79).

The robot trajectory from its starting point to the target point is visible in Figure 19. The target location is initially at coordinate (51, 79) on the right side of the user. The robot starts to head towards the target location but as soon as it gets close enough to the user, the particle filter gives more weight to the particles to the left of the user than to the right, in correspondence with the actual direction of approach of the robot. The choice of the robot final location is visible in Figure 20. The robot heads towards the left side of the user but the choice of the final destination is not only related to the proximity of the particles to the actual robot location. Indeed the final target location is at the left side of the user where there is the maximum of the preferred left side configuration (second row of Table 2).

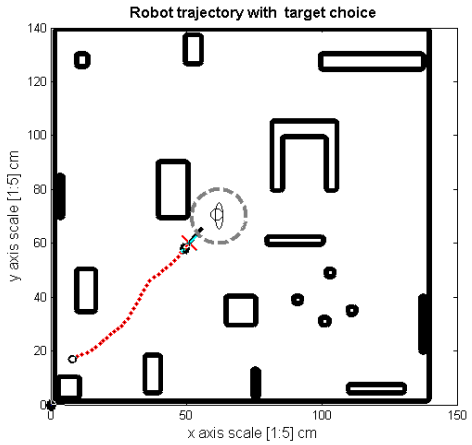

Figure 19: Diagram showing the trajectory of the robot from its initial point to the final point. The target is initially on the opposite side respect to its final location (cross). Since the robot is coming from the left the particle filter shifts the target to the left of the user (represented at the center of the slashed circle) leading to the selection of the final point (cross). The choice of the robot final location is mediated by the user preferences model expressed in Eq. (34).

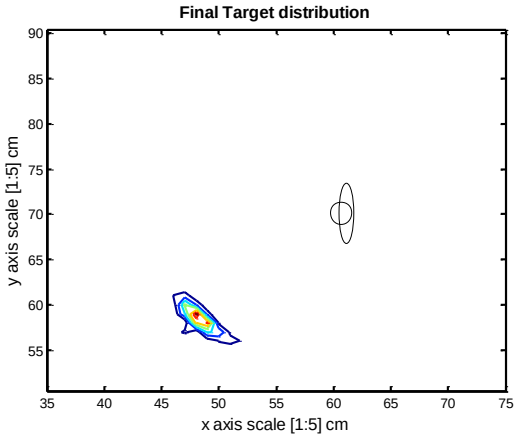

Figure 20: Diagram showing the final configuration of the target distribution. The particles are initially distributed around the user with a peak on the user right side. Since the robot is approaching the user from the left side, the particles are shifted towards the left and their final peak is at coordinate (51, 59), which gives also the maximum of the weighting function which parameters are expressed in the second row of Table 2.

6 Conclusion and discussion

In this deliverable we have presented the current state of the context awareness for robot navigation based on a real-time person tracking and robot navigation. The input to this system is a ceiling camera mounted in the centre of the KSERA lab, later to be mounted in the person's apartment. A hybrid probabilistic algorithm is described 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 tracking 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 on-line 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 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 the person.

A further task we are currently working on is to localise the robot in the room. This system can use robot-specific visual features such as the color, which does not change. A further challenge is that the robot orientation needs to be estimated. As long as this work has not been completed, we are using a large visual marker placed on the head of the Nao robot, however, that marker needs to be removed to give the robot its normal appearance and to enable the touch buttons, which are Nao's tactile sensors on his head.

For navigating the Nao robot inside the person's home, a behavior-based navigation system respecting the user’s personal space has been presented. A probabilistic density is calculated based on the modelling of the user’s preferred direction and the distance of approach. The validity of the behavior architecture and of the inference process have been tested in MATLAB™ [69] for generic robots and in Webots [75] for Aldebaran’s Nao [9]. The results of the simulations confirm the effectiveness of the model as well as the effectiveness of the inference process by means of the particle filter technique. The parameters of the model discussed in Section 4.2 have been derived from the HRI trials conducted by Walters [17] and Dautenhahn [47]. The behavior-based robot navigation system shows us robust and safe navigation ability in a cluttered and dynamically changing environment.

6.1 Discussion

6.1.1 Person localization

It may in the future be better if the system tracks a person not only based on the described visual cues, but also on some further features. Principally, the more cues there are, the better tracking performance could be achieved. Also, 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 person. We have experimented with a multilayer perceptron network based on moment invariant features [56] 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 leads only to

little improvement in distinguishing the person from the background. We are considering to include another person-specific cue in the future.

Though the initial aim 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. 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 [76].

6.1.2 Robot navigation

A particular emphasis of the robot navigation strategy in KSERA is the optimal placement of the NAO robot with respect to the user. The robot should not annoy the user. The existing studies about the optimal placement of robots w.r.t. humans were carried out with humanoid robots of comparable height as human beings. Thus, the model's parameters defining the human personal zones are not automatically valid for the NAO robot as it is only 60 cm tall. As a first approximation we assumed that the human personal zones still hold for the NAO, because a number of psychological experiments have been conducted to assess whether the height of the robot influences the optimal distance of approach, Syrdal et al. (RO-MAN 2007) and Walters et. al. [17]. According to their results the height of a robot has no substantial influence on the optimal distance of approach. Although their smallest robot was still 1,20m tall, we expect that NAO's limited height does not change the optimal distance very much. Nonetheless, the optimal model's parameters may be different for NAO as compared to other robots of human size. However, we expect this effect to be small and it does not change the structure of our model, nor does it affect its robustness much due to the probabilistic nature of our model The validation of the approach with a simulation leaves open the problematic related to the uncertainties in the readings of the user and robot poses. However, the representation of the user's proxemic space introduced by our model allows to constantly considering multiple hypothesis and thus coping with errors in the measurements readings. Future works will better characterize the model's performance in terms of robustness against uncertain sensor readings by means of experiments in the context of the prototype developments of WP 5.

References

[1] E. Aarts, R. Harwig, and M. Schuurmans, “Ambient intelligence, The invisible future: the seamless integration of technology into everyday life,” 2001.

[2] J. Nehmer, M. Becker, A. Karshmer, and R. Lamm, “Living assistance systems: an ambient intelligence approach,” in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 43–50.

[3] J. Hootman and C. Helmick, “Projections of US prevalence of arthritis and associated activity limitations,” *Arthritis & Rheumatism*, vol. 54, no. 1, pp. 226–229, 2006.

[4] 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)*. [Online]. Available: http://www.aal-europe.de/Published/reports-etc/-Final%20Version.pdf

[5] “OECD demographic and labour force database,” *Organisation for economic co-operation and development*, 2007. [Online]. Available: http://www.oecd.org/topicstatsportal/-0,3398,en_2825_494553_1_1_1_1_1,00.html

[6] M. Broxvall, M. Gritti, A. Saffiotti, B. Seo, and Y. Cho, “PEIS ecology: Integrating robots into smart environments,” in *Proceedings IEEE International Conference on Robotics and Automation, ICRA 2006.* IEEE, 2006, pp. 212–218.

[7] 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. http://www.ep.liu.se/ecp/048/008/, May 2010.

[8] Q. Meng and M. Lee, “Design issues for assistive robotics for the elderly,” *Advanced Engineering Informatics*, vol. 20, no. 2, pp. 171 – 186, 2006, engineering Informatics for Eco-Design. [Online]. Available: http://www.sciencedirect.com/science/article/B6X1X-4JMM07B-7/2/-18d10f94c0c26c9cb08b27b045f22986

[9] 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.

[10] 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.* IEEE, 2007, pp. 271–276.

[11] Y. Raoui, M. Goller, M. Devy, T. Kerscher, J. 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.* IEEE, 2009, pp. 1–6.

[12] T. Barger, D. Brown, and M. Alwan, “Health-status monitoring through analysis of behavioral patterns,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 1, pp. 22–27, 2004.

[13] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[14] H. Nait-Charif and S. McKenna, “Activity summarisation and fall detection in a supportive home environment,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, 2004, pp. 323–326.

[15] B. Zhang and H. Muhlenbein, “Synthesis of sigma-pi neural networks by the breeder genetic programming,” in *Proceedings of the First IEEE Conference on Evolutionary Computation*, Jun. 1994, pp. 318 –323 vol.1.

[16] S. Thrun, “Particle filters in robotics,” in *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, vol. 1, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/-download?doi=10.1.1.20.567&rep=rep1&type=ps

[17] M. Walters, K. Dautenhahn, R. Boekhorst, K. Koay, D. Syrdal, and C. Nehaniv, “An empirical framework for human-robot proxemics,” in *Proceedings New Frontiers in Human-Robot Interaction, K. Dautenhahn (Ed.), symposium at the AISB09 convention*. Citeseer, 2009, pp. 144–149.

[18] T. van Kasteren, G. Englebienne, and B. Kröse, “Activity recognition using semi-markov models on real world smart home datasets,” *Journal of Ambient Intelligence and Smart Environments*, vol. 2, pp. 311–325, 2010.

[19] B. Schiele, M. Andriluka, N. Majer, S. Roth, and C. Wojek, “Visual people detection – different models, comparison and discussion,” in *Proceedings of the IEEE ICRA 2009. Workshop on People Detection and Tracking*, 2009.
[20] G. Bauer and P. Lukowicz, *Computers Helping People with Special Needs*, ser. Lecture Notes in Computer Science, 2008, ch. Developing a Sub Room Level Indoor Location System for Wide Scale Deployment in Assisted Living Systems, pp. 1057–1064.
[21] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, “A joint particle filter for audio-visual speaker tracking,” in *Proceedings of the 7th international conference on multimodal interfaces*. ACM, 2005, pp. 61–68.
[22] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, “People tracking and following with mobile robot using an omnidirectional camera and a laser,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 557–562.
[23] R. Muñoz-Salinas, E. Aguirre, and M. Garcá-Silvente, “People detection and tracking using stereo vision and color,” *Image and Vision Computing*, vol. 25, no. 6, pp. 995–1007, 2007.
[24] S. Bahadori, L. Iocchi, G. Leone, D. Nardi, and L. Scozzafava, “Real-time people localization and tracking through fixed stereo vision,” *Applied Intelligence*, vol. 26, no. 2, pp. 83–97, 2007.
[25] 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,” *J Multimodal User Interfaces*, vol. 2, no. 2, pp. 75–91, 2008.
[26] O. Lanz and R. Brunelli, “An appearance-based particle filter for visual tracking in smart rooms,” in *Classification of Events, Activities and Relationships - CLEAR*, 2007, pp. 57–69.
[27] 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*, vol. 56, no. 10, pp. 857–864, 2008.
[28] G. West, C. Newman, and S. Greenhill, *From Smart Homes to Smart Care*, 2005, ch. Using a Camera to Implement Virtual Sensors in a Smart House, pp. 83–90.
[29] M. Piccardi, “Background subtraction techniques: a review,” in *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 2004, pp. 3099 – 3104 vol.4.
[30] I. Jolliffe, *Principal Component Analysis*. John Wiley & Sons, Ltd, 2005. [Online]. Available: http://dx.doi.org/10.1002/0470013192.bsa501
[31] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
[32] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[33] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” vol. 3951, pp. 404–417, 2006.
[34] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.
[35] 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*, vol. 2, no. 1, pp. 53–62, 2010. [Online]. Available: http://www.iai.uni-bonn.de/~frintrop/paper/-frintrop_etal_soro2010.pdf
[36] F. Hecht, P. Azad, and R. Dillmann, “Markerless human motion tracking with a flexible model and appearance learning,” in *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, May 2009, pp. 3173 –3179.
[37] D. Ramanan, D. Forsyth, and A. Zisserman, “Tracking people by learning their appearance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 65–81, 2007.
[38] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition CVPR 2000 Cat NoPR00662*, vol. 2, no. 8. IEEE Comput. Soc, 2000, pp. 142–149.
[39] Z. Zivkovic and B. Krose, “An EM-like algorithm for color-histogram-based object tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004.*, vol. 1, 2004.

[40] J. Triesch and C. Malsburg, “Democratic integration: Self-organized integration of adaptive cues,” *Neural Computation*, vol. 13, no. 9, pp. 2049–2074, 2001. [Online]. Available: http://-citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.1337&rep=rep1&type=pdf
[41] S. M. Khan and M. Shah, “Tracking multiple occluding people by localizing on multiple scene planes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 505–519, 2009.
[42] Y. Qian, G. Medioni, and I. Cohen, “Multiple target tracking using spatio-temporal markov chain monte carlo data association,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE, 2007, pp. 1–8.
[43] K. Smith, D. Gatica-Perez, and J. Odobez, “Using particles to track varying numbers of interacting people,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 962–969, 2005.
[44] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. Christensen, “Navigation for human-robot interaction tasks,” in *Proceedings. ICRA ’04. 2004 IEEE International Conference on Robotics and Automation*, vol. 2, 26-may1 2004, pp. 1894 – 1900 Vol.2.
[45] Y. Nakauchi and R. Simmons, “A social robot that stands in line,” *Autonomous Robots*, vol. 12, pp. 313–324, 2002, 10.1023/A:1015273816637. [Online]. Available: http://dx.doi.org/-10.1023/A:1015273816637
[46] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, “A model of proximity control for information-presenting robots,” *Robotics, IEEE Transactions on*, vol. 26, no. 1, pp. 187 –195, 2010.
[47] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami, and T. Siméon, “How may i serve you?: a robot companion approaching a seated person in a helping context,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, ser. HRI ’06. New York, NY, USA: ACM, 2006, pp. 172–179. [Online]. Available: http://doi.acm.org/-10.1145/1121241.1121272
[48] A. Oskoei, M. Walters, and K. Dautenhahn, “An Autonomous Proxemic System for a Mobile Companion Robot.” AISB, 2010.
[49] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” *The International Journal of Robotics Research*, vol. 25, no. 12, p. 1243, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.9030&rep=rep1&type=pdf
[50] L. Kaelbling, M. Littman, and A. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.9565&rep=rep1&type=pdf
[51] 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*, vol. 20, no. 11, pp. 1254–1259, 1998.
[52] K. Bernardin, T. Gehrig, and R. Stiefelhagen, “Multi-level particle filter fusion of features and cues for audio-visual person tracking,” *Multimodal Technologies for Perception of Humans*, pp. 70–81, 2009.
[53] A. Khotanzad and J. Lu, “Classification of invariant image representations using a neural network,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 6, pp. 1028–1038, 2002.
[54] M. Swain and D. Ballard, “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
[55] M. Mercimek, K. Gulez, and T. Mumcu, “Real object recognition using moment invariants,” *Sadhana*, vol. 30, no. 6, pp. 765–775, 2005.
[56] M. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 2002.
[57] S. Pal and S. Mitra, “Multilayer perceptron, fuzzy sets, and classification,” *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 2002.
[58] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” *Cognitive modeling*, p. 213, 2002.
[59] J. Czyz, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1271–1281, 2007.

[60] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” *Computer Vision CCV 2002*, pp. 661–675, 2002.
[61] S. Sural, G. Qian, and S. Pramanik, “Segmentation and histogram generation using the HSV color space for image retrieval,” *Proceedings International Conference on Image Processing. 2002.*, vol. Vol 2, 2002.
[62] R. Arkin, *Behavior-based robotics*. The MIT Press, 1998.
[63] E. Bicho, “The dynamic approach to behavior-based robotics,” 2000.
[64] R. Soares, E. Bicho, T. Machado, and W. Erlhagen, “Object transportation by multiple mobile robots controlled by attractor dynamics: theory and implementation,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 292007-nov.2 2007, pp. 937 – 944.
[65] G. Schöner, M. Dose, and C. Engels, “Dynamics of behavior: Theory and applications for autonomous robot architectures,” *Robotics and Autonomous Systems*, vol. 16, no. 2-4, pp. 213 – 245, 1995, moving the Frontiers between Robotics and Biology. [Online]. Available: http://-www.sciencedirect.com/science/article/B6V16-3Y5FR82-8/2/b58c2ba87d8c996a76149423bbf91fcc
[66] E. Hall and E. Hall, *The hidden dimension*. Doubleday Garden City, 1966, vol. 6.
[67] R. Kirby, R. Simmons, and J. Forlizzi, “Companion: A constraint-optimizing method for person-acceptable navigation,” in *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, 272009-oct.2 2009, pp. 607 –612.
[68] D. Lambert and D. Group, *Body language*. HarperCollins, 2004.
[69] Matlab. http://www.mathworks.com/.
[70] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon, “A human aware mobile robot motion planner,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874 –883, 2007.
[71] E. Pacchierotti, H. Christensen, and P. Jensfelt, “Evaluation of passing distance for social robots,” in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, 2006, pp. 315 –320.
[72] S. Thrun, “Probabilistic robotics,” *Commun. ACM*, vol. 45, pp. 52–57, March 2002. [Online]. Available: http://doi.acm.org/10.1145/504729.504754
[73] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 25, no. 7, pp. 53 –82, 2010.
[74] B. Keni and S. Rainer, “Evaluating multiple object tracking performance: the CLEAR MOT metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, p. 10, 2008.
[75] O. Michel, “Cyberbotics Ltd. WebotsTM: Professional mobile robot simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
[76] P. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, no. 4, p. 711, 1995.