A Graphical Model for Unifying Tracking and Classification within a Multimodal Human-Robot Interaction Scenario

Tobias Rehrl, Jürgen Gast, Nikolaus Theißing, Alexander Bannat,
Dejan Arsić, Frank Wallhoff, Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
http://www.mmk.ei.tum.de

Christoph Mayer, Bernd Radig
Chair for Image Understanding and Knowledge-Based Systems
Technische Universität München
http://www9.in.tum.de

Abstract

This paper introduces our research platform for enabling a multimodal Human-Robot Interaction scenario as well as our research vision: approaching problems in a holistic way to realize this scenario. However, in this paper the main focus is laid on the image processing domain, where our vision has been realized by combining particle tracking and Dynamic Bayesian Network classification in a unified Graphical Model. This combination allows for enhancing the tracking process by an adaptive motion model realized via a Dynamic Bayesian Network modeling several motion classes. The Graphical Model provides a direct integration of the classification step in the tracking process. First promising results show the potential of the approach.

1. Introduction

Due to the fact that human-robot interaction (HRI) has an enormous impact on ongoing research efforts, several research facilities pursue the goal of establishing a natural and intuitive HRI. In general, the information gained from image processing techniques contribute a major part to the knowledge required for HRI, nonetheless additional information channels like audio-signals can improve the interaction as well. Vision-based data can provide information about the human (position, gestures, etc.) as well as information about the environment (objects, scene understanding, etc.). Audio-based data can be utilized to establish a dialog between the human and the robot as well as to gain knowledge about the scene (classifying environmental noises, sound-localization, etc.).

HRI covers many areas of image and audio processing, thus, we limited our research focus to the following topics: gesture recognition, and human-machine dialogs. Despite this limited selection, we attempt to process entire data in a unified framework and envision to process all occurring problems in a holistic way to realize a multimodal HRI.

The rest of this paper is organized as follows: A brief overview of related work concerning the topics covered in this paper is presented in Section 2. In Section 3, we introduce our scenario and the robotic platform. The multimodal capable processing framework is presented in Section 4. In Section 5, the Graphical Model for combining tracking and classification is delineated, in addition first promising results are shown. The paper closes with a summary and an outlook over the next planned steps.

2. Related Work

Ambient Assisted Living [15] is a research area, where the integration of robotic platforms in the interaction with humans obtains more and more importance. In [8] a mobile robotic research platform is presented, which assists humans in their daily life. In [1] the objective of the project is a combination of robotics and ambient intelligence technologies. This is mediated by a mobile robotic companion working in a smart home environment. In general, all research activities attempting to establish a natural HRI tackle this problem by following a multimodal strategy, like in [21] speech recognition and visual perception techniques are integrated on a humanoid robot for accomplishing this goal.

Gestures are classical nonverbal means of communication, thus, several approaches [10, 12] can be found to integrate gestures in the HRI to achieve a natural and intu-

978-1-4244-7028-0/10/$26.00 ©2010 IEEE

itive form of communication. For showing agreement or disagreement, head gestures are an intuitive way, however, like in [17] they were applied for controlling operations (document browsing, dialog box confirmation). In addition to head gestures, hand gestures are applied as a nonverbal communication form, e.g. in [11] a system is presented capable to recognize hand gestures for a HRI context. A particular form of hand gestures is given by pointing gestures, see [18] for an example. Pointing gestures can provide information for the localization of objects and persons, when a person is pointing at them to indicate the position of the object or of the person for a robot.

The approach presented in this paper deals with the combination of particle tracking and Dynamic Bayesian Network classification. To the best of the authors' knowledge, there has been little effort made to combine tracking and classification in the way presented here, using Graphical Model-based inference and prediction as the dynamical model of the particle filter. However, some approaches exist heading in a similar direction.

A motion-based particle filter for head tracking was proposed in [4]. It utilized adaptive Block-Matching as a dynamical model, thus improving the linear process model while still avoiding the necessity of offline learning. The analytical justification for its superiority over the standard Condensation tracking [13] was given in [5].

In [9], a combination of the Condensation algorithm and Graphical Models for tracking and classification of the tracked features was presented. Here, the tracking process was used to acquire the feature sequences of the observed motion. After the tracking, the feature sequences were used as an input for a Graphical Model-inference-based classification.

A combination of Condensation algorithm and Graphical Models in order to improve the tracking was presented in [22], where facial expressions were observed. In that approach, the temporal progression was subjected to the linear process of the particle filter, whereas the spatial correlation between the facial features was inferred by an undirected Graphical Model in each frame to enhance the tracking results.

An integration of stochastic states into the dynamical model of a particle filter was introduced in [19]. Here, a Condensation algorithm was implemented which utilized several Gaussian Auto-Regressive Processes as possible dynamical models. The decision about which of them to use for particle propagation was made by a finite-state machine describing the several motion classes along with their transition probabilities.

A memory-based particle filter was proposed in [16], this approach stores the previous estimated states and utilizes them to generate a prior distribution. From the previous states a probability is calculated indicating the likelihood of a past state to appear in the future again. Similar to our approach, this approach does not apply the Markov assumption anymore and it is capable of handling non-linear, time-variant, and non-Markov dynamics. Our approach, in contrast, applies learned motion patterns via a Dynamic Bayesian Network, whereas the memory-based particle filter uses the history of the past states to provide estimates for the future.

3. Scenario and Research Platform

The Scenario is situated in an Ambient Assisted Living environment. A human interacts with the robot ELIAS to get information (e.g., what room a person is sitting in, when the next train is leaving, etc.), to start services (listen to music, make a video phone call, etc.) or to get a desired object (in our case, by the help of other robots performing the manipulation task). The focus of the research within the scenario is to make the interaction with the mobile robotic platform more natural by exploiting different communication channels, like speech and nonverbal information (gaze, head or hand gestures, facial expressions). A typical interaction situation is depicted in Figure 1.

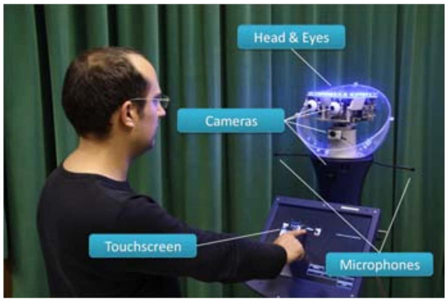


Figure 1. Human browsing through information displayed on the touchscreen of the mobile robotic research platform.

The robot (a commercial research platform) has several sensors to monitor the interaction environment. Besides a laser scanner and sonar sensors for the navigation, two microphones capture the audio-signals for the subsequent speech processing. Multiple cameras are installed to detect persons, faces, gestures, and objects. Furthermore, the platform is equipped with a head with two degrees of freedom and two eyes movable with high speed. The computing power of the research platform is constituted by two PC's (an industrial one running Linux used for image processing and system control and a MAC-Mini for displaying the graphical user interface, speech recognition and synthesis). The currently implemented features of the platform are a knowledge-based system controller, modules for speech input and output, a recognizer for head gestures to confirm or abort a speech command, an attention system based on the users head pose and certain hand gestures. For the person identification a face identification algorithm using Hidden

Markov Models and Eigenfaces is applied. Furthermore, a badge recognizer is implemented to extract the bearer's name from his badge. Finally, a face model is used to provide information about the users emotions (joy, neutral, etc.) and micro-mimics (e.g. lifting of eyebrows).

4. Framework

This section delivers an overview about the applied system architecture. The knowledge based system controller is using facts and rules. Perception modules update or provide their observations in form of facts. The rules are used to reason about the next steps and thereby trigger the action modules. These perception and action modules can connect to the system controller via two middlewares. One middleware is the Internet Communication Engine (ICE) simplifying distributed computing including different operating systems and programming languages.

The second applied middleware is a local sensory buffer using shared memory, which is called Real-time Database (RTDB). The RTDB origins from research about a communication framework for cognitive autonomous vehicles (see [7]) and is designed to deal with vast amounts of data streams captured at different data rates. The basic input for the RTDB is provided by modules writing their sensor information into a shared memory and label the data with a timestamp. For each data type a container has to be defined (e.g., image data is stored in the IplImage_struct of OpenCV [6], allowing to reuse the already existing algorithms). The RTDB-Manager allocates a ring buffer for the data container when the data is created for the first time. This allows reading modules not only to access the most recent data, but also to look back in time, depending on the available shared memory and the update rate of the sensor. Furthermore, this shared memory allows multiple modules to access the same data (e.g. a camera image) without blocking effects. The processing modules write back their resulting information (e.g. the position of a face) and thereby provide them for the next modules in the processing queue. The modules can be controlled (start, stop, pause, resume) via the RTDB to manage and distribute the available computational power among the modules required within the current context. However, the timestamps of the data allow to align the multimodal data of the different sensors and modules for further processing. The derived high level information (e.g. a gesture) is passed to the system controller to decide the next actions. The framework is in use on another service robot platform as well as on an industrial robotic platform, where a human and a robotic assistant work in a hybrid assembly scenario. Further input modalities comprise gaze information of the user, a photonic mixer device delivering depth information and physiological data (pulse, skin conductance, heart beat).

5. Research Focus

As mentioned above, the general goal is to accomplish a multimodal HRI, therefore, we highlight our major research vision, a holistic approach for handling problems in HRI. The general idea behind this vision is the integration and the establishment of interrelations between modules of different origin (feature extraction, classification, dialog unit, etc.) in a unifying form, for which the described framework provides a good foundation. For the computer vision-based processing, we combined the tracking and classification processes in a unifying Graphical Model (GM). Before we introduce our new approach, we outline briefly the basic concepts (GMs, particle tracking).

5.1. Basic Concepts

5.1.1 Graphical Models

Graphical Models (GMs) [14] are applied in many areas of research, since they provide a descriptive and illustrative way to depict problems regarding control theory, computer science, pattern recognition, etc. In general, the GMs combine probability theory and graph theory portraying the interdependences between different random variables. In this paper we consider only directed GMs, also known as Bayesian Networks (BNs). When BNs model time series data, they are called Dynamic Bayesian Networks (DBNs), which we applied to model the dynamical model of the particle filter. Hidden Markov Models (HMMs) are a sub-class of DBNs, where observations are dependent on an unobservable variable, referred as *hidden state*.

An efficient inference algorithm for GMs is the *Junction Tree Algorithm*. This algorithm uses cluster potentials (cliques ψ and separators ϕ) for describing the dependencies between random variables $(X_1, \dots, X_n)$ by the quotient of cluster and separator potentials

$$p(X_1, \dots, X_n) = \frac{\prod_{C \in \mathcal{C}} \psi(C)}{\prod_{S \in \mathcal{S}} \phi(S)}. \tag{1}$$

The DBN modeling the dynamical models of the different motion classes was realized with the Graphical Model Toolkit [3].

5.1.2 Particle Tracking

Reliable tracking of objects in a video sequence is still a challenging task for current research activities. The Condensation algorithm [13] is a robust tracking method successful even under unfavorable conditions.

The observed sequence $\mathcal{Z}_T = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$ is related to the desired information, which is referred to as the state sequence $\mathcal{X}_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ of the pattern.

In each frame t, the state $\mathbf{x}_t$ of the observed object influences the observation $\mathbf{z}_t$, which is therefore exclusively dependent on $\mathbf{x}_t$:

$$p(\mathcal{Z}_t|\mathcal{X}_t) = \prod_{i=1}^{t} p(\mathbf{z}_i|\mathbf{x}_i). \tag{2}$$

Additionally, in the classical Condensation algorithm the object states $\mathbf{x}_t$ are assumed to be subject to the Markov property, i.e. dependent only on their immediate temporal predecessor:

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{3}$$

The density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ represents the dependency of the current state on its predecessor, i.e. the change of the state vector over time. Thus, the term can be interpreted as the dynamical model describing the motion of object.

The Condensation algorithm can be subdivided into two steps: prediction of the current state $\mathbf{x}_t$ having the observed sequence $\mathcal{Z}_{t-1}$, and a measurement step having the entire observation sequence $\mathcal{Z}_t$. Combining both steps and regarding equation 2 and the fact that given a state $\mathbf{x}_t$ the observation sequence $\mathcal{Z}_t$ has no more information, the recursive step from one frame to its successor is given by:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = k_t p(\mathbf{z}_t|\mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})\, d\mathbf{x}_{t-1}. \tag{4}$$

with $k_t = 1/p(\mathbf{z}_t|\mathcal{Z}_{t-1})$.

A sampling with the Monte Carlo Method is used to approximate the computationally infeasible integration over the state $\mathbf{x}_{t-1}$.

5.2. Combining Tracking and Classification

The approach concerning tracking and classification is the attempt to combine both methods in a unifying Graphical Model (GM-approach), where both methods can profit from each other. Therefore, a particle filter is used for the tracking part, whereas the classification is performed by a DBN. The interface between tracking and classification is the dynamical model in the particle filter: The term $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ serves as an estimation of the object motion, approximated with a linear filter. If this dynamical behavior is, instead, computed using inference on a DBN describing the dynamics more exactly, both tracking and classification can be improved. An *adaptive motion model* can be used to adapt the tracking process according to a set of motion classes $\mathcal{M}$. The motion class is determined via a DBN, according to that result the dynamical model of the tracking process can be adapted, forming a closed *tracking-classification-tracking* loop. In addition to the adaptive motion model, a *classification step* can be performed by classifying the entire observed sequence.

5.2.1 The Graphical Model

The process generating the observed feature vector sequence of a gesture class c with a certain motion pattern m is modeled by the GM in Figure 2. The prolog and the $T-1$ chunks constituting the GM consist of five nodes:

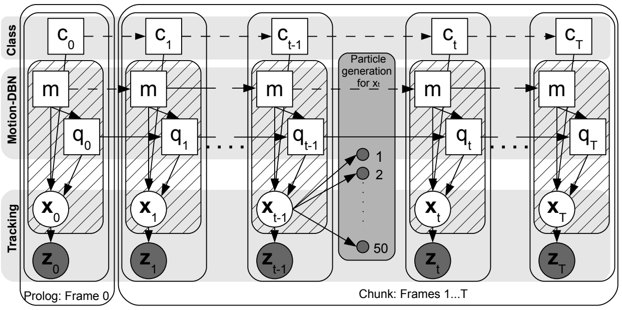


Figure 2. GM-approach for combining tracking and classification.

The *gesture class* c_t represents the gesture expressed via a certain motion pattern. The gesture class c_t is appointed to a motion sequence of the entire observation range T. An observation sequence is assumed to correspond to one gesture from its beginning to its end. The gesture class hence remains unchanged throughout the sequence.

The *motion class* m_t represents the motion class of the observed pattern indicating the kind of motion an observed object performs. The transitions between adjacent m_{t-1} and m_t are deterministic in the way that m_t copies the state of m_{t-1}. The reason is that an observation sequence is assumed to consist of one complete motion, from its beginning to its end. The class of motion hence remains unchanged throughout the sequence.

The *temporal progression state* q_t resembles the hidden state variable in a HMM. It depicts the temporal progression of the motion as each state describes a discrete time step.

The *state vector* $\mathbf{x}_t$ denotes, in this case, the position of the tracked object.

The observation vector $\mathbf{z}_t$ is the actual observation. In this application, it is an array of image pixels.

The objective of this tracking and classification approach is twofold: For the *adaptive motion model*, the objective is to deduce for the observed sequence $\mathcal{Z}_{t-1}$ the most probable motion class m_{t-1}. With this information, the particles are distributed according to the transition of temporal progression state q_{t-1} to q_t to infer the best prediction for the next state $\mathbf{x}_t$. For the classification step, the objective is to deduce the most probable gesture class $\hat{c} \in \mathcal{C}$ for the entire observation sequence $\mathcal{Z}_T$.

5.2.2 Adaptive Motion Model

As above mentioned, the combination of tracking and classification is achieved fusing both in a unifying GM via the dynamical model of the particle filter expressed by $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. This first-order process in equation 4 is replaced here by inference in a DBN taking all other RVs into account, thus the Markov property is not valid anymore constituting an important difference between our *adaptive motion model*-approach and the classical Condensation.

In the prediction step, the a-priori-density is created as

$$p(\mathcal{X}_t|\mathcal{Z}_{t-1}) = p(\mathcal{X}_{t-1}|\mathcal{Z}_{t-1})p(\mathbf{x}_t|\mathcal{X}_{t-1}), \qquad (5)$$

since with given state vectors in the past, the past observation vectors do not provide any new information, thus $p(\mathbf{x}_t|\mathcal{X}_{t-1}, \mathcal{Z}_{t-1}) = p(\mathbf{x}_t|\mathcal{X}_{t-1})$.

Using the independence of an observation $\mathbf{z}_t$ on any RV except of its corresponding state $\mathbf{x}_t$ yielding $(\mathbf{z}_t|\mathcal{X}_t, \mathcal{Z}_{t-1}) = p(\mathbf{z}_t|\mathbf{x}_t)$, the a-posteriori-density in the measurement step can be expressed as

$$p(\mathcal{X}_t|\mathcal{Z}_t) = k_t\, p(\mathbf{z}_t|\mathbf{x}_t)p(\mathcal{X}_t|\mathcal{Z}_{t-1}), \qquad (6)$$

with $k_t = 1/p(\mathbf{z}_t|\mathcal{Z}_{t-1})$, and where $p(\mathbf{z}_t|\mathbf{x}_t)$ can be evaluated by computing the value of a weight function.

$p(\mathbf{x}_t|\mathcal{X}_{t-1})$ is the prediction of the state vector in frame t from the state vectors in frames $1 \ldots t-1$. This term is evaluated via Bayesian inference by marginalizing over each RV in the path between $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ (see Motion-DBN in Figure 2):

For inference and prediction, only the crosshatched subgraph of the GM in Figure 2 is used which does not contain the observation vectors $\mathbf{z}_t$. From $\mathcal{X}_{t-1}$ inference is applied to predict the next state vector x_t by creating 50 particles utilizing the learned transition probabilities from the Motion-DBN and measure their correlation with the observation $\mathbf{z}_t$.

The motion class m is assumed to remain constant throughout the observed motion, i.e. $m_t =: m$ for any timestep t. With this, the probability function term simplifies to

$$p(m_t|m_{t-1}) = \delta(m_t, m_{t-1}), \qquad (7)$$

with the Kronecker delta δ.

Regarding this property the prediction term is given by

$$\begin{aligned} p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \quad & \textstyle\sum_m \sum_{q_{t-1}} \sum_{q_t} p(m, q_{t-1}|\mathcal{X}_{t-1}) \\ & p(q_t|m, q_{t-1})p(\mathbf{x}_t|m, q_t). \qquad (8) \end{aligned}$$

By splitting the leftmost term and shifting the innermost sum according to the distributive law, the conditional probability mass function

$$\begin{aligned} p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \quad & \textstyle\sum_m p(m|\mathcal{X}_{t-1}) \sum_{q_{t-1}} p(q_{t-1}|m, \mathcal{X}_{t-1}) \\ & \textstyle\sum_{q_t} p(q_t|m, q_{t-1})p(\mathbf{x}_t|m, q_t) \qquad (9) \end{aligned}$$

describes which operations have to be performed by an algorithm in order to calculate $p(\mathbf{x}_t|\mathcal{X}_{t-1})$:

For each possible motion class m and motion state q_{t-1} in the previous time-step, their respective probabilities have to be determined from the terms $p(m|\mathcal{X}_{t-1})$ and $p(m, q_{t-1}|\mathcal{X}_{t-1})$, given the knowlegde of all previous state vectors $\mathcal{X}_{t-1}$. Then, the transition probability to each current motion state q_t is calculated by $p(q_t|m, q_{t-1})$ from its predecessor. Finally, the term $p(\mathbf{x}_t|m, q_t)$ predicts the current state vector $\mathbf{x}_t$ as a multi-dimensional mixture of Gaussian components whose parameters are learned in advance. This density function provides the Gaussian means and covariances for each tracked state, while the other densities can be seen as weighting factors. Thus in each time step the algorithm samples from each motion class, from each preceding state and each current state. Using these values, it tracks the pattern state by sampling from a weighted set of Gaussian curves.

Tracking-Classification-Tracking Loop After the initialization of the tracker, in this case, we assumed the knowledge about the location of the first state $x_{t=0}$, the *tracking-classification-tracking* loop starts. The following steps are performed until the end of the sequence $t = T$ is reached:

Variable prediction: The motion class m_t and motion state q_t of the current frame are sampled with knowledge of the particle sequence.

Particle prediction: The state vector x_t has to be predicted by the sampled movement class and current state.

Measurement: verification of the validity of the predicted sample by measurement.

Resampling: The new particles are sampled out of the set of preceding samples considering the result of a weight function.

Inference: applying Bayesian inference from Section 5.2.2 to determine $p(\mathbf{x}_t|\mathcal{X}_{t-1})$.

5.2.3 Classification Step

Besides the *adaptive motion model*, the combination of tracking and classification can be used to deduce a gesture class $\hat{c} \in \mathcal{C}$ from the entire observation sequence $\mathcal{Z}_T$.

The set of all discrete observation steps is denoted as $\mathcal{Z}_T = \{z_1, \ldots, z_T\}$, where T is the number of discrete time steps. The gesture class of interest is given by

$$\hat{c} = \arg\max_c p(c|\mathcal{Z}_T). \qquad (10)$$

The observation model employed in the Condensation algorithm assumes the observation $\mathcal{Z}_T$ to be a cluttered, noise-affected origin of the actual, hidden state sequence $\mathcal{X}_T$ which describes all relevant properties of the pattern. A marginalization over the set of all possible states $\mathcal{X}_T$ yields

$$p(c|\mathcal{Z}_T) = \int_{\mathcal{X}_T} p(c|\mathcal{X}_T)p(\mathcal{X}_T|\mathcal{Z}_T)\, d\mathcal{X}_T \qquad (11)$$

with $p(c|\mathcal{X}_T, \mathcal{Z}_T) = p(c|\mathcal{X}_T)$, since according to the observation model, $\mathcal{Z}_T$ does not contain any additional information given $\mathcal{X}_T$.

Inserting this into (10) yields

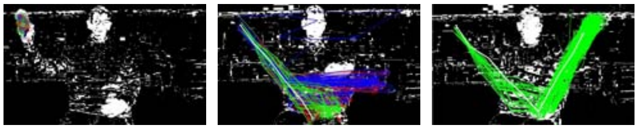
$$\hat{c} = \arg\max_c \int_{\mathcal{X}_T} p(c|\mathcal{X}_T) p(\mathcal{X}_T|\mathcal{Z}_T)\, d\mathcal{X}_T. \qquad (12)$$

With all respective conditional probability functions replaced accordingly and any constant terms removed from the $\max_c$ statement, the gesture class $\hat{c}$ is expressed as

$$\begin{aligned}\hat{c} = \quad & \arg\max_c \int_{\mathbf{x}_1} \cdots \int_{\mathbf{x}_T} \prod_{1\leq t\leq T} p(\mathbf{x}_t|\mathcal{X}_{t-1}) p(\mathbf{z}_t|\mathbf{x}_t) \\ & \cdot p(c|\mathcal{X}_T)\, d\mathcal{X}_T. \qquad (13)\end{aligned}$$

5.2.4 Experiments

We tested the system with simulated and real-recorded data sequences (ten sequences per class for simulated data, and eleven sequences per class for real data) for three different classes (left to right, vee move, zorro move). The simulated sequences were generated by a moving white rectangle in a noisy environment formed by white circles and some randomly distributed rectangles. The real-recorded data sequences were skin-color image sequences (obtained by applying a method like presented in [18]) of a test person performing the three gestures. One gesture sequence is depicted in Figure 3. For the initialization a fixed number of 50 particles were used. The particles were set on the relevant object, i.e. human hand, or moving rectangle. In order to compare our GM-approach, the obtained results for the tracking and classification are compared to those of a reference model, a Condensation tracking using Brownian Particle Motion (Brownian-approach) [20]: Gaussian Normal Distribution with mean value $\mu = 0$, and standard deviation $\sigma = 100$px to predict the particles' dynamic behavior.



(a) 1. frame (b) 6. frame (c) 10. frame

Figure 3. In the 1. frame, the particles are placed on the hand (initialization) for the three gestures: Vee move (green), Zorro move (blue), and left to right (red). For the 6. and 10. frame, the results of the Motion-DBN and their related tracking paths are indicated with their corresponding colors. The most probable path is indicated in white.

Tracking Results For evaluating the tracking performance in comparison to existing methods, the measures Tracker Detection Rate (TRDR) and Object Tracking Error (OTE) presented in [2] are applied. In each frame, the centroid of the tracked object is calculated and compared to that of the Ground Truth data. The average Cartesian distance between them in each frame constitutes the OTE. The TRDR is determined by defining a distance threshold and checking whether each single distance between the centroids falls below it. Since in this application the object of interest (the hand) has a diameter of approximately $d = 140$ px, the threshold is defined as $d/2 = 70$px. Each frame with a distance lower than $d/2$ is rated as a positive detection. In Table 1, the obtained tracking results for the TRDR and OTE for the GM-approach and the Brownian-approach are shown.

	GM-approach	Brownian-approach
OTE	47.9 px	132.4 px
TRDR	77.8%	33.5%

Table 1. Tracking performance results.

Classification results The 63 gesture sequences (three per class) were classified by using the Motion-DBN, the results for the two approaches (GM-based, Brownian-based) can be seen in Table 2 in form of confusion matrices.

	GM-approach			Brownian-approach		
	leftright	vee	zorro	leftright	vee	zorro
leftright	19	0	2	0	0	21
vee	0	19	2	0	0	21
zorro	0	0	21	0	0	21

Table 2. Classification results.

Interpretation In general, the results of the presented GM-approach are significantly better than the tracking results of Brownian-approach. The weak performance of the Brownian-approach is due to the frequent clutter in the data sets and the Brownian Motion dynamical model which should provide a generic dynamical model. Thus, the performance of the reference model can be clearly improved by utilizing a better dynamical model. In addition, the current GM-approach lacks of the capability to track a motion pattern clearly different from the motion classes modeled by the Motion-DBN.

The classification results of the reference model (in this case the Brownian-approach) showed a weak performance, since the tracking process failed and thus the classification. In this case, the zorro gesture is some kind of garbage class for the Brownian-approach.

5.2.5 Discussion

The current approach for combining tracking and classification in a unifying GM demonstrates much potential, however, there are many possibilities left to proof the potential of the approach and to optimize it.

First, the current gesture set is too limited and the motion patterns show a clear difference, thus, the DBN-based classification can perform quite well. Therefore, the gesture set should be increased by gestures having a quite overlap in their motion patterns. Besides, gestures having non-linear motion patterns can demonstrate the capability of the presented approach to handle these kinds of patterns. In addition, the gesture set (three gestures, ten sequences per class for simulated data, and eleven sequences per class for real data) can be extended in the number of gestures and for the real data sequences in the number of test persons.

Second, the current approach lacks real-time capability and thus can presently not be integrated on the robotic platform, however, it is imaginable to use the Compute Unified Device Architecture (CUDA) to speed up the the Bayesian inference for the particles. Due to design of the DBN-based classification for the motion classes for each particle, there is room for parallelizing the operations and process them via the graphic card.

6. Conclusion and Future Work

A new approach for combining tracking and classification in a unifying GM was presented. The approach can improve the tracking via an adaptive motion model, and the classification can be integrated in the tracking process. First evaluations show the potential of the approach, however, there is still room for improvement left. First, the current system lacks of real-time capability, since the bottle-neck is the integration of the GMTK-based classification results. Second, the number of motion class can be extended and the number of particles can be reduced until a optimal amount is obtained. Third, the interrelation between the gesture class and the motion class can be varied that several gestures can be modeled with a limited amount of motion patterns.

References

[1] A. Badii. Companionable - integrated cognitive assistive & domotic companion robotic systems for ability & security. 2009. 1

[2] F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. In *In PETS*, 2006. 6

[3] J. Bilmes. Gmtk: The graphical models toolkit, 2002. 3

[4] N. Bouaynaya and D. Schonfeld. A complete system for head tracking using motion-based particle filter and randomly perturbed active contour. volume 5685, pages 864–873. SPIE, 2005. 2

[5] N. Bouaynaya and D. Schonfeld. On the optimality of motion-based particle filtering. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19(7):1068–1072, 2009. 2

[6] G. Bradski and A. Kaehler. In *Learning OpenCV: Computer Vision with the OpenCV Library*. O'ReillyPress, 2008. 3

[7] M. Goebl and G. Färber. A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles. *Intelligent Vehicles Symposium*, pages 737 – 740, June 2007. 3

[8] B. Graf, C. Parlitz, and M. Hägele. Robotic home assistant care-o-bot®3 product vision and innovation platform. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II*, pages 312–320, Berlin, Heidelberg, 2009. Springer-Verlag. 1

[9] R. Hamid, Y. Huang, and I. Essa. Argmode - activity recognition using graphical models, 2003. 2

[10] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno. Knowledge-based person-centric human-robot interaction using facial and hand gestures. volume 3, pages 2121 – 2127 Vol. 3, oct. 2005. 1

[11] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno. Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform. *Robot. Auton. Syst.*, 2007. 2

[12] T. Hashiyama, K. Sada, M. Iwata, and S. Tano. Controlling an entertainment robot through intuitive gestures. volume 3, pages 1909 –1914, oct. 2006. 1

[13] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998. 2, 3

[14] M. I. Jordan, editor. *Learning in graphical models*. MIT Press, Cambridge, MA, USA, 1999. 3

[15] M. C. Katrin Gaßner. Ict enabled independent living for elderly – a status-quo analysis on products and the research landscape in the field of ambient assisted living (aal) in eu-27. Technical report, Institute for Innovation and Technology, 2010. 1

[16] D. Mikami, K. Otsuka, and J. Yamato. Memory-based particle filter for face pose tracking robust under complex dynamics. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:999–1006, 2009. 2

[17] L.-P. Morency and T. Darrell. Head gesture recognition in intelligent interfaces: the role of context in improving recognition. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 32–38, 2006. 2

[18] K. Nickel and R. Stiefelhagen. Visual recognition of pointing gestures for human–robot interaction. pages 1875–1884. Elsevier, 2007. 2, 6

[19] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1016–1034, 2000. 2

[20] M. Smoluchowski. Zur kinetischen theorie der brownschen molekularbewegung und der suspensionen. *Annalen der Physik*, (21):756–780, 1906. 6

[21] R. Stiefelhagen, H. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel. Enabling multimodal human robot interaction for the karlsruhe humanoid robot. *Robotics, IEEE Transactions on*, 23(5):840 –851, oct. 2007. 1

[22] C. Su and L. Huang. Spatio-temporal graphical-model-based multiple facial feature tracking. 2005(13):2091–2100, 2005. 2