FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D2.2005
Joint RA2 Deliverable
--
First algorithms for robust human body tracking. Strategies for combination / fusion of different tracking methods. Classification and first methods for recognition of composite human activities.

Due date of deliverable: 31/12/2005
Actual submission date: 25/01/2006

Start date of project: January 1st, 2004 **Duration** : 48 months

Contributing partners :
UniKarl, LAAS, UniBi, UvA, IPA, UH

Revision: final
Dissemination level: PU

Executive Summary

The work conducted as part of the RA2 *Detection and Understanding of Human Activities* during the second phase (month 12 – month 24) was organized in three workpackages:

WP2.1 Detection and perception of body parts based on sensor features

WP2.2 Human body model: Integration and fusion

WP2.3 Context based interpretation and classification of activities

These workpackages are closely linked together along an overall process chain, which is depicted in Fig. 1. It consists of a 3 level architecture, which enables the integration of different tracking algorithms, and the fusion of results into an abstracted 3D human model, which serves as input for the activity recognition. Beside the modularity, the strengths of the approach lie in the fact that tracking algorithms of different complexity and granularity can be fused together. The fusion is guided by a confidence measure, which has to be provided by each tracker. Reciprocally, the tracker granularity can be specified by the algorithms of the high level through quality parameters. The whole process consists of two streams, namely the information flow downwards and a control parameterization upwards.

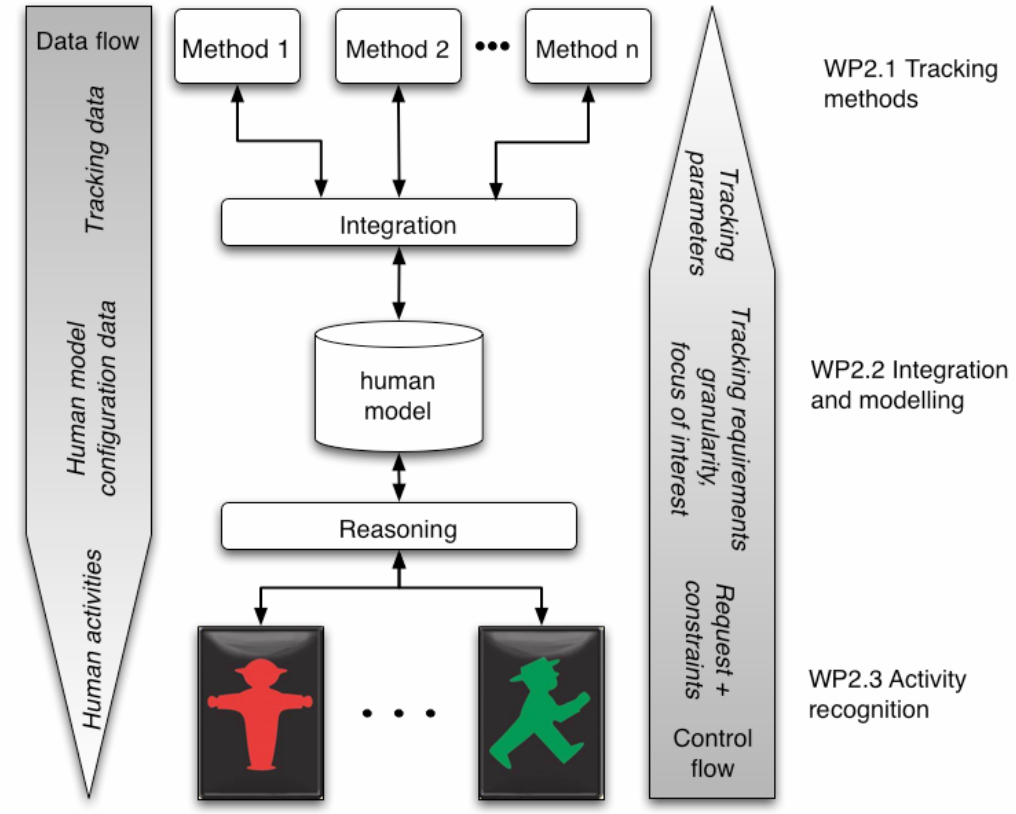


Fig. 1 Connections and relations between the different workpackages in RA2

Within the second phase, some effort was spent in order to improve the developed tracking methods with respect to robustness and computational cost. Further, in the context of WP2.2 a fusion framework, able to fusion 2D and 3D data has been implemented. In parallel, complementary fusion methods based on Particle Filtering were implemented as part of the developed 2D tracker. For WP2.3, a set of activities has been derived in cooperation with RA3, and a classification mechanism has been developed, which is able to distinguish a large set of activity classes. Also within WP2.3, an object attention system has been set up and implemented. It enables the robot during a dialogue to resolve references to objects and spaces that are currently in focus by a combination of verbal information, gesture recognition and color-based object models.

The report is structured in 3 sections according to the workpackages. Each section starts with an outline of the performed work and the presentation of the role of the partners. Next, the work description and the achieved results are presented along referenced publications. Finally, an outline concerning the future work is given.

Role of RA2 in Cogniron

The main goal of RA2, the detection and understanding of human activities, denotes one of the basic components of a robot acting in close cooperation with humans. This capability builds the basis for or is at least strongly linked to all other components of a "cognitive robot" like a multi-modal dialog system (RA1), interaction and social behaviour (RA3), learning through demonstration (RA4), multi-modal situation awareness and spatial cognition (RA5) and finally for deciding whether to take initiative (RA6).

RA2 has a close link to RA1. This is formed via WP2.3, where the Object Attention System (OAS) has been developed and will be extended. This OAS is an integral part of the dialogue system, which is developed within RA1.

RA2 and RA3 also collaborate closely, with WP2.3 and WP3.4 working together in a loop. Within RA3, occurring human activities are derived from user studies, which are then implemented in RA2. Additional input on occurring features for these activities will be provided by RA3, while RA2 in turn supports the user studies which are carried out by RA3 with the developed tracking systems to give feedback both to the user and to the developers.

RA4 and RA6 both benefit from the results obtained in RA2: For learning and imitation (RA4) as well as for physical interaction and decision making (RA6) it is crucial to have information on the human body pose and the currently performed activity.

To obtain information about objects and places, which is needed as additional input ("extrinsic features") for activity recognition as well as for the Object Attention System, RA2 relies on RA5. Here, RA2 incorporates results from object and space modelling.

Relation to the Key Experiments

Results of RA2 will be used for all Key Experiments (KEs). In KE1 (Home Tour Scenario), functionalities for human body tracking (Cogniron function CF-PTA), for 3D gesture recognition (CF-GR), for resolving object references (CF-ROR) and for detection of human activities (CF-ACT) will be especially demonstrated. Tracking functionality will be provided by WP2.1 and WP2.2, while recognition and interpretation of human gestures and activities results from WP2.2 and WP2.3.
KE2 (*The Curious Robot*) focuses also on human-robot collaboration. Results on recognition, interpretation and understanding of human actions and activities (CF-ACT) from WP2.3 will be integrated there. For haptic interaction like passing objects information about the human body configuration is crucial and will be provided by CF-TBP within WP2.1/WP2.2.
The ability to observe human actions is also requisite for KE3 (*Learning skills and tasks*), which focuses on learning from observation of humans. Humans performing e.g. manipulation tasks have to be observed in order to reason about the performed task. So the functionality of tracking body parts (CF-TBP) containing trajectories characterizing the performed motion will be part of KE3.

Contents

1 Detection and perception of body parts based on sensor features (WP2.1)

1.1 Introduction

The objectives of the work done during the second project year in WP2.1 were the development, the implementation, as well as further improvements of tracking methods for 2D and 3D tracking of humans and human limbs. For both cases different tracking methods were implemented in order to cope with requirements like accuracy, robustness, reliability and computational time.

Fig. 2 shows how the different parts of the developed methods work together, respectively what information is transferred between them. Note that the fusion of tracking results is done in WP2.2, but even on the level of tracking methods the exchange between different modules is enhancing the result of every single method. 3D tracking is based on a Time of Flight sensor and is needed for gathering the trajectories of limbs and the whole body configuration. Since the resolution of the sensor is not fine enough, finger and hand movements can't be tracked. Therefore, for close and mean range 2D vision tracking is used in order to track hands (configuration or pose) or the head (direction). For a robust tracking a good initialisation respectively a coarse position estimation of body parts to track is advantageous. Such information combined with the identification of humans at long distances is provided by another 2D vision based tracker taking into account the movement of the robot. Further plausibility measures are used to estimate the quality of the tracking.

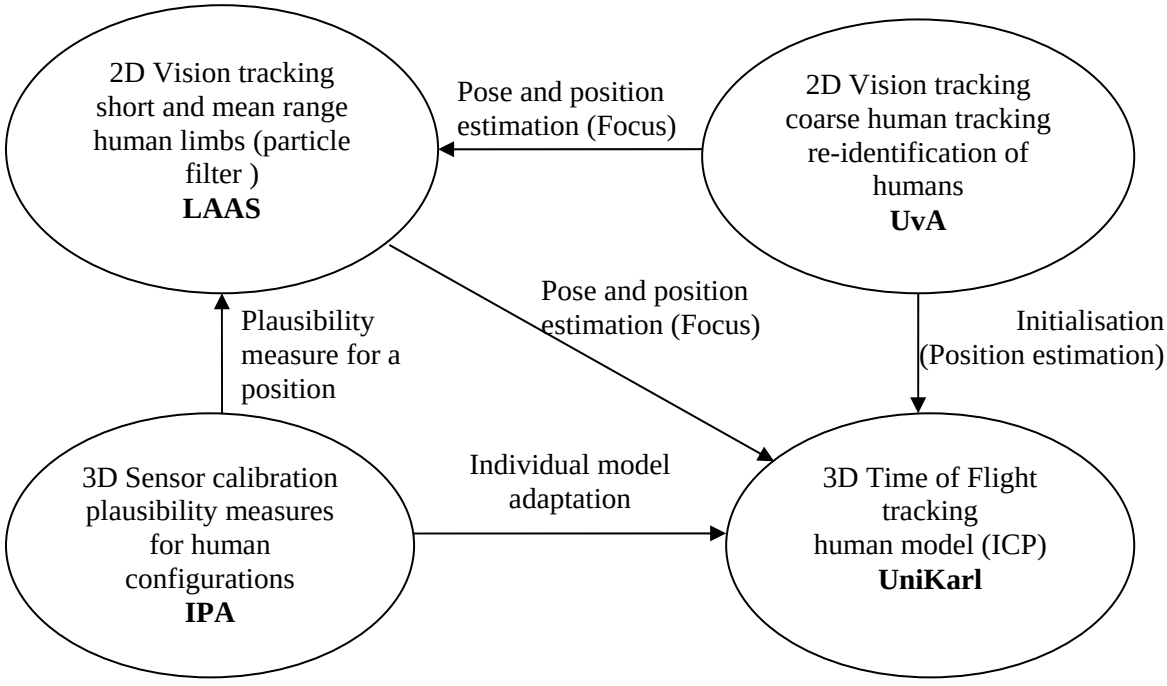


Fig. 2 Organisation and information flow between the developed methods.

1.2 Work description

For the coarse human tracking the work was done on improving the video based human tracking and detection methods developed in the first phase. It is assumed that there are initially a few persons in the clear space in front of the robot and that they can be reliably detected using a simple state of the art

algorithm, for example, motion segmentation as we did in the first phase. The histogram based representation from the first phase was used to model and track the person. We developed a Bayesian framework where the histogram based representation can be combined with different image cues. We tested the robustness of different Bayesian filtering schemes for person tracking using color histogram based features combined with image motion features. For details please refer to [1, 4]. Furthermore we prepared experiments to collect and annotate video and other sensor data in targeted the scenario of person following the robot (and other way around).

For 2D tracking of humans and body parts at several ranges, a particle filtering framework was developed [2], which enables the design of 2D visual trackers fusing/combining in a principled manner information that comes from various measurement sources. Although this fact has been acknowledged before, it has not been fully exploited in the robotics context. Several such 2D visual trackers were implemented, relying on various visual cues (based on all or parts of colour, shape, motion) and diverse particle filtering schemes. The aim was to increase the versatility and reliability of 2D trackers dedicated to human limbs for three Human-Robot (H/R) interaction modalities: (i) short-range head tracking for proximal H/R distance, (ii) human tracking for mean-range H/R distance, (iii) monitoring for long H/R distance.

For the 3D human tracking, the work addressed in the past year concentrated on the implementation of fast and robust methods. The aim was to enable the Iterative Closest Point (ICP) based fusion and tracking framework to track a person with more than 10 Hz frame rate and cope with natural limbs and body movements. For this the used articulated 3D human body model was enhanced by introducing new joint models, which represent the human joint constraints in terms of degrees of freedom. The joints between the cylinders of the human representation are modelled as artificial point correspondences, which results in a set of forces and torques maintaining the model constraints. (For details on the used joint models refer to section 5 of [3].) By doing so, the constraints are an immanent part of the ICP based tracking, which directly leads the human configuration iteratively to a next valid one. Finally with the aid of improvements on the calibration of the 3D sensor (Time of Flight) and optimizations of the software implementation, the goal of robust tracking was reached.

1.3 Results

For coarse human tracking [1, 4] a large set of surveillance data is used to test the robustness of different Bayesian filtering schemes for the person tracking using colour histogram-based features combined with image motion features. The method is evaluated by measuring the number of misidentified observations (identification error) and the error in the number of distinct people recognized from the data. The persons were observed in two places with similar illumination conditions. During a second experiment, the platform travelled a longer distance, between two places where the illumination conditions changed significantly. In this case one out of five encountered persons was wrongly identified as a new individual (rather than a previously observed one). The complete set of labelled clips is available at the following address: http://www.science.uva.nl/wzajdel/Icra05.

The experiments simulate home-like environments, where the robot has to cope with a small number of distinct persons. The tests show, that although the robots postulates a new person with every new local trajectory, the model is able to estimate the number of distinct people when the illumination changes slowly. The results shows that the various multiple hypothesis filtering schemes perform similarly on the dataset. The Kalman doesn’t fulfil this requirement as it maintains only a single hypothesis. Another conclusion is that the robustness to varying light conditions still needs to be improved.

For 2D short and mean range tracking [2] the particle likelihoods defined from the visual cues, together with the importance functions specified from visual detectors, were thoroughly evaluated off-line on images grabbed in the robotics context, in terms of discriminative power, precision and time consumption. Concerning the estimation engine, the standard particle filtering formalism as well as alternative schemes were evaluated in order to check which ones best fulfils the requirements of the considered H/R interaction modalities. The evaluated strategies are CONDENSATION, ICONDENSATION, the Auxiliary Particle Filter, the Rao-Blackwellized Subspace History-Sampling Sampling Importance Resampling (RBSSHSSIR) algorithm and the Hierarchical Particle Filter [1].

The performance of the trackers, as well as their robustness w.r.t. illumination and appearance changes, occlusions and jumps in the dynamics, were assessed on many representative sequences in the context of the three H/R interaction modalities mentioned above. For each of these, a separate visual tracking strategy was thus selected, namely, the *head tracking modality* combines motion and shape cues into a CONDENSATION algorithm, the *human tracking modality* merges shape and colour information into a RBSSHSSIR algorithm, while the *monitoring modality* relies on color and motion distributions into an RBSSHSSIR algorithm.

The selected trackers were implemented on the Rackham robot for KE2. The whole evaluation is gathered in the PhD thesisof Ludovic Brèthes (defended on 2005/12/13) [8], and will soon be submitted for publication. An excerpt [5] of some results of the thesis, which are presented on http://www.laas.fr/~lbrethes/KE2_2k5/, is appended to this document.

The 3D-ICP based tracking approach [3] does not include any background knowledge apart from kinematic constraints, i.e. no assumptions like “the torso stands always upright” are made. This implies on the one hand that all possible body configurations can be recognized; on the other hand, the tracking can succeed only if the input data contains all necessary information to determine the human posture, and no tracking hypothesis can be generated for temporarily invisible body parts or ambiguous configurations.

The current frame rate is approx. 10-14 Hz. The computation time depends on several factors: It scales linearly with the number of measured 3D points on the model; background points are removed in an early stage and do not distinctly influence the needed time. It also depends on the number of ICP steps performed in each frame, which is approx. 3-15, depending on the desired accuracy and the speed of the movement.

The evaluation step consisted in recording a set of 100 sequences which contain ten different movements from several persons: e.g. point somewhere, walk, wave, shake hands with somebody, bow or clap. The tracking results have then been evaluated and classified manually into one of three classes: (0) Tracking lost somewhere within the sequence, (1) acceptable deviations like a temporally lost (but recovered) forearm within a walking sequence, and (2) good congruence between original and resulting model movements. The evaluation result showed that only in 5% of the sequences the tracking had lost the subject completely and for 32% minor temporal deviations had occurred.

Different conclusions can be drawn from the results. Huge movements are easily detected by the 3D data based tracking: The “bow” movement is tracked quite well. On the other hand, fast movements with the extremities may cause failures when only 3D data is used, as with the “wave” movement. The experiments also have shown that tracking based only on the measurements of the Time of Flight camera is not sufficient. Especially movements along the main axis of the body (e.g. sitting down) can hardly be detected, which substantiates the use of different data inputs for a fusion algorithm, which is described in WP2.2, and also the results derived from the fusion of 2D and 3D tracking.

[1] The CONDENSATION and ICONDENSATION algorithms were formerly developed by Blake and Isard in 1996 and 1998, respectively, the Auxiliary Particle Filter was defined by Pitt and Shephard in 1998, the RBSSHSSIR algorithm was developed by Torma and Szepesvári in 2003, and the Hierarchical Particle Filter is an extension of the Partitioned Particle Filter (Mac Cormick and Isard 2002) developed by Pérez et al. in 2004.

1.4 Future work

For the future, the existing detection, identification and tracking methods will be further improved. Especially the initialisation and plausibility checks will be integrated in the tracking algorithms.

Concerning coarse tracking, we plan to extend the current features by additional visual cues and to use different invariant representations. Handling of partial occlusions will be approached by explicitly modelling occlusions of different parts of the coarse human model. Furthermore, the geometric constraints and motion models for human and robot including robot odometry should guide the initialization and tracking. Finally, in collaboration with RA3, we will investigate the scenario of robot following human and vice versa. This analysis will be used to develop appropriate, - mainly vision based- perception methods for this special situation, which is especially applicable for KE1.

Concerning future developments of the 2D short and mean range tracking, work will be done concerning the introduction of auditory cues to vision-based human tracking, based on the information delivered by a wideband sound source localization system under development at LAAS[2].

The focus of the 3D tracking will lay on the observation of manipulation actions, were a major point is the tracking of the hand and the grasped object. In the framework of the ICP tracking simplified object models will be integrated into the human model by extending the kinematic chain. This work will be done in close collaboration with RA5 were object recognition and learning of these models is investigated.

1.5 References

[1] W.Zajdel, Z.Zivkovic and B.Krose, ***Keeping track of humans: have I seen this person before?***, ICRA 2005, Barcelona, Spain, 2005.

[2] Ludovic Brethes, Frédéric Lerasle, Patrick Danès, ***Data fusion for visual tracking dedicated to human-robot interaction*** , ICRA 2005

[3] Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann, ***Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP***, International Conference on Humanoid Robots (Humanoids 2005), Tsukuba , Japan

[4] Z. Zivkovic, A.T. Cemgil, B. Krose, ***Approximate Bayesian methods for kernel-based object tracking***, IEEE Transactions on Pattern Analysis and Machine Intelligence, (journal paper under revision).

[5] Excerpt from http://www.laas.fr/~lbrethes/KE2_2k5/

[6] S. Argentieri, P. Danès, P. Souères, **Prototyping Filter-Sum Beamformers for Sound Source Localization in Mobile Robotics,** ICRA'05

[7] Sylvain Argentieri, Patrick Danès, Philippe Souères, and Pierre Lacroix, ***An Experimental Testbed for Sound Source Localization with Mobile Robots using Optimized Wideband Beamformers***, IROS'05

[8] Ludovic Brèthes. **Visual Tracking using Particle Filters. Application to H/R interaction**. PhD Thesis, LAAS-CNRS and Paul Sabatier University, December 2005.

[2] Some preliminary work on sound source localisation has been performed and presented in [6, 7]

2 Human body model: Integration and fusion (WP2.2)

2.1 Introduction

In the context of the WP2.2, fusion of information cues was implemented on different levels. Two different fusion aspects were focused: first, fusion of sensor and feature cues within the frameworks of the different tracking methods (2D with different ranges and 3D) and second, global fusion of the tracking results into a 3D human body model. From the point of view of an integrative human model the two fusion aspects can be viewed as local (first case) and global fusion methods (second case). The border between the two cases is not strict and will be part of future investigations. Especially the fusion frameworks based on Particle Filtering and ICP will be focused in the next period in order to gather a very good solution for the separation problem.

In Fig. 3 the role and contribution of the partner involved in WP2.2 is presented. The developed local fusion methods are obviously closely related to the developed tracking algorithms in WP2.1.

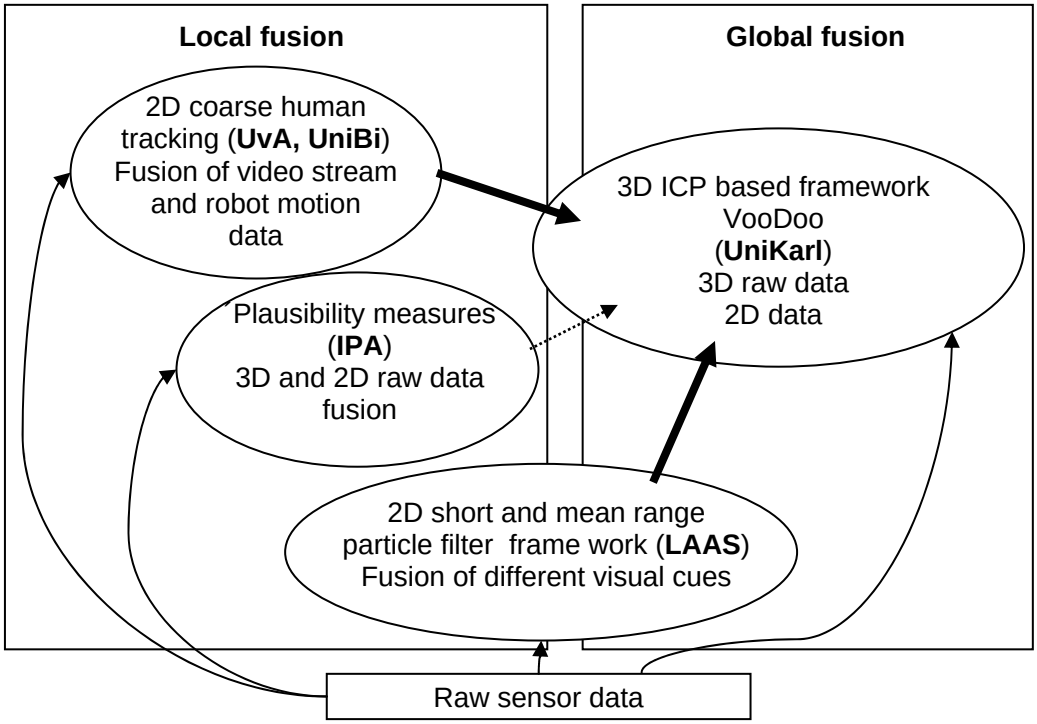


Fig. 3 Organisation and information flow.

2.2 Work description

During the past year, an ICP based fusion and tracking framework named VooDoo [1] was developed. It follows a new approach for fusion of different input sensors and cues for tracking immanently into a 3D human body model. This approach is able to incorporate tracking information from 3D sensors like Time-of-Flight-cameras (ToF) or stereo reconstruction, together with cues from 2D trackers entailing a single camera.. The system is designed to work solely with sensors on-board the robot. It can perform tracking and fusion in real-time at about 10-14 Hz.

The main features of the VooDoo fusion method are:

- real-time immanent fusion of different tracker and sensors inputs
- hierarchical model of body and body parts

- fusion based on confidence measures for every input cue
- integration of plausibility check for body configurations
- adjustable tracking granularity on body part level
- output of confidence measures on body part level

In Fig. 4 the structure of the Voodoo algorithm is presented. The algorithm is based and guided by the ICP iteration (main loop), which incorporates different modules. Fusion of all data is done on the 3D point level, according to the working principle of the ICP. 2D information, e.g. from a monocular system is mapped to a ray into the 3D space and associated to the corresponding body part. 3D information is inserted depending on the correspondence level into the different modules. All points are weighted according to the confidence measures, by increasing the point weight[3]. The resulting confidence measure is computed from the associated / used normalized number of points corresponding to the body parts.

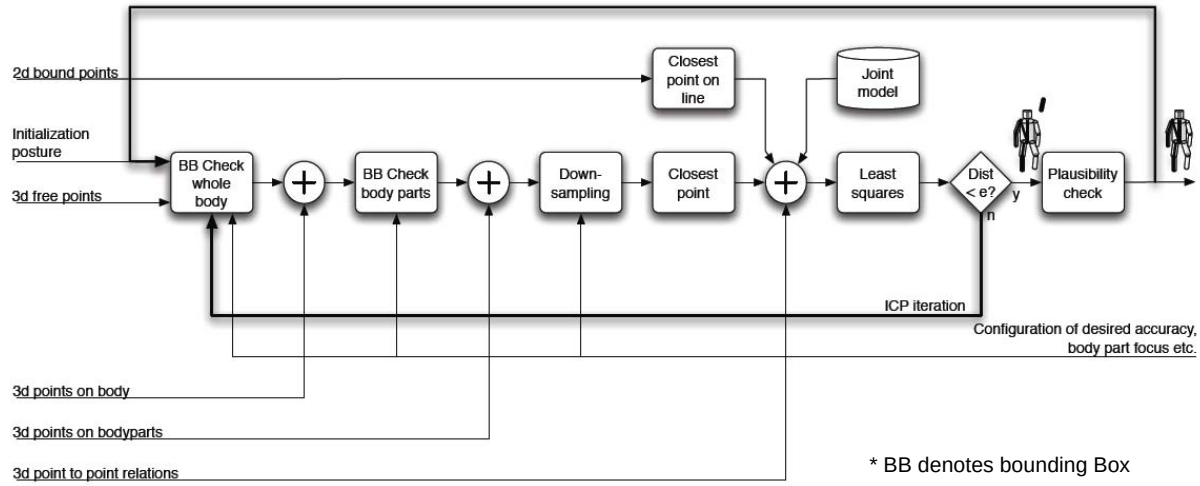


Fig. 4 Structure of the VooDoo algorithm [2].

The *Plausibility Checker* function tests the results of the human tracking algorithm in order to determine the actual validity of a current configuration. A data base of rules and facts has been set up according to medical literature [4], which allows distinguishing between enfant, children, youngster, and adults. At start the plausibility checker function compares ratios of head/body size and shoulder width/body size with predefined values. If the ratios are correct, further tests will be initialized based on the category of the detected person. The checking is split in two phases: checking of the length of each limb and checking the joint angles of the limbs.

Some preliminary work was done in [2] regarding an appearance-based approach for 3D tracker of human limbs. It concerned the fusion of 2D features in order to estimate the configuration of a 3DOF arm by Unscented Kalman filtering. This appearance-based 3D tracker will be extended to all the upper human limbs, based on the combination/fusion of visual cues developed during the 2nd phase of the project together with information coming from 3D sensor (TOF or stereoscopic system), and considering alternate stochastic estimation schemes. This work will be continued during the following project phases, where some effort will be devoted to increase the overall precision and robustness and to achieve satisfactory computational performance.

Besides the work on local fusion method in the particle filtering framework in the context of 2D tracking already presented in section 1.2, the fusion of visual and acoustic cues will enable 2D person

[3] Increased point weight does not influence the runtime of the ICP loop.

tracking without a laser-range finder [3]. This allows reliable tracking of communication partners and also constitutes a valuable input cue for 3D fusion.

2.3 Results

The evaluation of the fusion capability of the VooDoo framework was done by using 3D point clouds and the corresponding color images from 100 sequences of movements of different subjects. The evaluation was done separately by testing the tracking with 3D and 2D data only and finally using both input cues. The 3D data has been acquired with the Time of Flight camera and the 2D data is derived from skin color segmentation in one image of the stereo camera. The extracted skin blobs were associated to the hands and the head of the tracked persons. For fusion the corresponding lines (integration of 2D information in the 3D space) were connected to the endpoints of the forearm cylinders respectively the head centre. For the results presented in [1], the following weights for the input data have been used: 3D data points w = 1.0, face tracker w = 30.0, hand tracker w = 20.0.

Comparing the results of the isolated tracker and the fused one, it can be seen in [1] that the later overcome the weaknesses of the single algorithms. For example a “bow” movement is not tracked correctly by the 2D tracker, and on the other hand the “wave” movement is not followed correctly by the 3D tracker. All used sequences can be found on http://wwwiaim.ira.uka.de/users/knoop/cogniron. This test set will be used for further evaluation and data sharing.

The introduction of the 2D hand and head tracking cues does not influence the frame rate of the VooDoo fusion algorithm. As stated earlier, the frame rate depends on the number of different 3D points, and that means that either higher weights of points have no negative effect and also a constant number of lines (three in the experiments) from the 2D tracker can be neglected for the cost estimation of the algorithm. Therefore, the frame rate remains on 10-14Hz using approx. 3 – 15 iteration steps for one frame. Through the integration of different trackers the number of iteration steps will be decreased, due to a better estimation of corresponding body parts.

Concerning the results for fusion of several upper limbs within the particle framework a conceptual proof was done. The extension of the 3D tracker to all the upper human limbs planned for the next phase will give a reliable estimation of the potential of the method. The outcome of that work will serve as basis for the comparison of the two fusion frameworks and consequently enable to decide how to design the interface between them, respectively define the separation between local and global methods.

2.4 Future work

The status of the implementation of the VooDoo framework in terms of methodology and interface is nearly complete and a first evaluation performed. However, the integration of all tracking algorithms presented in WP2.1 has to be done next and hereby the adaptation in form of development of mapping strategies for the confidence measurements will be the focus of the further work in this workpackage. Additional work on model parameter estimation from key poses will be carried out, in order to improve the initialisation of the algorithm.

Another topic of work will be the comparison and evaluation of the fusion schemes (ICP versus Particle Filter based methods), as well as the derivation of feasible tracking quality measures.

Future work will be dedicated to fusion of appearance and visual cues for 3D human tracking purpose.

2.5 References

[1] S. Knoop, S. Vacek and R. Dillmann, ***Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model***, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006), May 15, Orlando, Florida. (accepted for publication)

[2] Mathias Fontmarty, ***Towards 3D tracking of a human limbs using several cameras for H/R interaction,*** Master thesis in Control, Computer and Decisional Systems, August 2005, LAAS & National Institute of Applied Sciences, Toulouse.

[3] T. Spexard, J. Fritsch, M. Kleinehagenbrock, and G. Sagerer, ***Human-like Person Tracking with an Anthropomorphic Robot,*** Proceedings of the IEEE International Conference on Robotics and Automation, 2006, to appear.

[4] Brandenburgische Umweltberichte (BUB), ***Köpermaße 2000: aktuelle Perzentilwerte der deutschen Bevölkerung im jungen Erwachsenenalter***. 2001.

3 Context based interpretation and classification of activities (WP 2.3)

3.1 Introduction

The objectives of WP2.3 during the second year stressed four aspects of human activity interpretation and classification (Fig. 5). As the first aspect (user) studies on gesture and activity classification were performed in close collaboration with RA3, RA1 and RA6 in order to identify gesture and activity classes and possible discriminating features. The outcome is passed to the recognition modules. The second aspect denotes the classification of communicative and commanding gestures based on 2D tracking information. The third focuses on the implementation of a classification framework for basic human activity recognition.

The fourth aspect tackles as part of WP2.3 was dedicated to the interpretation of pointing gestures in terms of dereferencing of objects. As underlying work, the focus of attention of the system had to be determined and controlled. This work is done in collaboration with RA5 (object learning and recognition) and RA1 (specification of features through dialog), and has a broad interface to RA3 providing the intention of a pointing gesture.

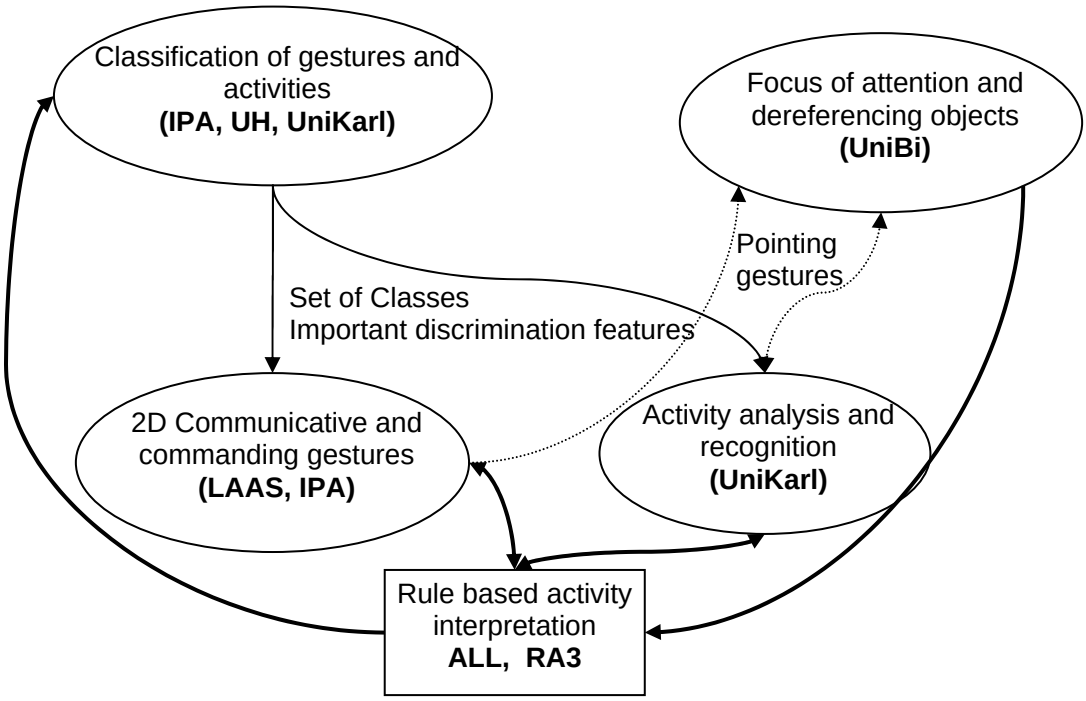


Fig. 5 Structure of performed work and the role of the involved partner

3.2 Work description

For the classification of gestures a methodological approach for identification of the human intent behind a gesture is discussed in [3]. The classification is intended as a generally applicable basis for incorporating the understanding of gestures into human-robot interaction. In order to infer intent from gesture, a broad classification of types of gestures into five main classes is introduced. The classification stresses the fact that for robot interaction there is a strong need to take into account not only the kinematics, but also the interactional context. To each of the five distinguished classes, the defining characteristics and associated intents are given. Further, some requirements of the operational classification and initial steps for its deployment are discussed. This work has been performed in collaboration with RA3 and was extended by recent user studies on manipulative activities. The

current work and the planned user studies on various situations related to the Key Experiments will give a refinement of the classification and the corresponding defining characteristics.

Concerning 2D communicative gestures interpretation, the approach [4] has been devoted to the classification of hand configurations (depending on the number of open fingers) and of fronto-parallel motions in the video stream. A mixed-stated CONDENSATION has been proposed to detect the most likely hand posture and canonical motion model, thus ensuring an automatic switch between multiple templates/motions in the tracking loop. The aim is to get a compact representation of a gesture through a sequencing of hand configurations and motions.

For the recognition of activities (CF-ACT) the analysis of the movements of the 3D human body model was addressed. The recognition approach is based on the hybrid activity hierarchy, which has been developed during the first project phase. An overview of the developed structures can be found in [1]. For this project phase, some of the activities were tackled, which have been derived from the current KE specification and through user studies in RA3. The approach is based on the assumption that each activity can be modelled by a set of characteristic features, which describe motion primitives of body parts, relations between body parts and objects, or relations between the motion/configuration and the observer. The activity recognition was performed by Feed Forward Neural Network (FFNN) classifiers.

A feature based approach for activity recognition was proposed. The human motion capture system gathers data of the human configuration over time. From this data different features are extracted. Features can be derived from raw data (like the position of a limb), from statistical analysis (like PCA or FFT) or from external modules (like objects in vicinity). For each activity a subset of features is extracted using *Mutual Information Feature Selection (MIFS)*, which selects the most relevant features for the given activity. The set of selected features is then used as input for the FFNN classifier. For each activity a FFNN has been trained with manually segmented training examples. The whole process is depicted in Fig. 6.

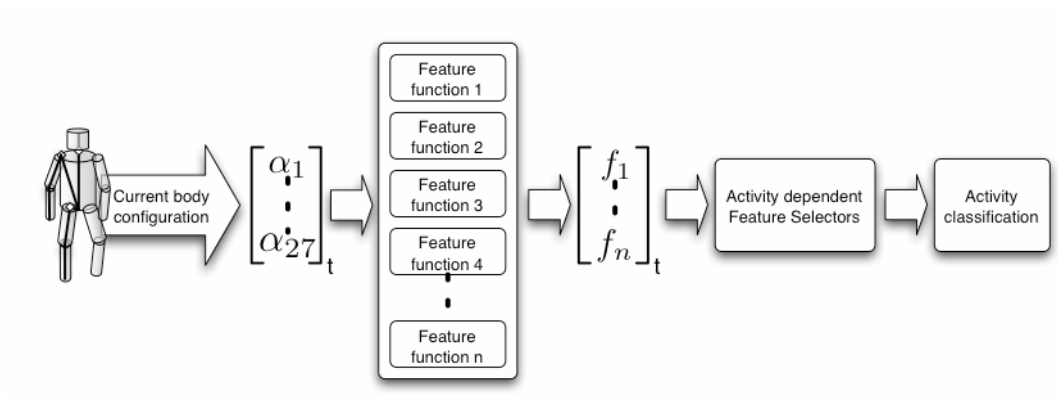


Fig. 6 Overall recognition process

During the second phase an Object Attention System (OAS) [5,6] has been developed that can resolve multi-modal references to objects. For this task the recognition results from a pointing gesture recognition system are combined with information from the dialog developed in RA1. If a multi-modal reference is given by the dialog, the OAS queries the gesture recognition for a potential pointing gesture. If available, the OAS focuses the pan-tilt camera of the robot on the position the user pointed at in order to acquire visual information about the corresponding object (e.g., view, position). If ambiguities arise in this process (e.g., more or less than one object found) the OAS can ask the dialog system to clarify the problem with the user [7]. At the end of this interaction, the acquired data is stored in the so-called scene model that currently contains only information about learned objects.

This includes additional information given verbally by the user like, for example, the owner of an object.

For the object detection process, a crucial distinction is made between object types which are known to the robot and those which first have to be learned. In order to ascertain whether an object type is known or not, a description of the object given by the user (e.g., type, color, owner) is sent to the scene model. If it is known, the scene model returns all object entries matching the symbolic description of the specified object. In order to verify if one of the returned objects is indeed the object the user refers to, the OAS has to analyze the camera image. This can be done by feeding previously learned object models from the scene model to an object recognition algorithm. In the current implementation, a simple appearance-based object recognizer is used for testing the OAS. In the future we plan to apply object recognition approaches developed in RA5 for solving the object recognition task.

If the object type is unknown, the camera image is searched for verbally specified visual features. Currently we apply color-based attention filters where the link between a verbally specified color like `blue' and the color values for the attention filter is realized with a modality converter. If an image region matching the search criteria is found, then it is supposed to contain a view of the referred object and is stored in the scene model. If the given verbal information is insufficient to determine a view, the user is asked through the dialog for more object features.

3.3 Results

As a conclusion of the discussion in [3] on how to infer the intent of a human interaction partner, a classification of gesture according to some major types in five tentative classes is proposed. The intent may be (1) absent / directed to objects or environment, (2) incidentally expressive, (3) symbolic, (4) interactional, or (5) deictic. A summary of the classes is given by Table I in [3]. For detecting the distinctions between the (sometimes overlapping) classes, the use of knowledge of human activity, recognition of objects and persons in the environment, and previous interactions with particular humans, as well as knowledge of conventional human gestural referencing and expression, in addition to specialized signalling codes or symbolic systems is needed.

For the decisional abilities of a robot, a task oriented process is needed. Gestures of type 3 and many of type 1 can be considered as task-oriented and the inference of their intent can be done relative to the task at hand. Gestures of type 4 include generic interactional gestures that may serve to manage the session itself: inviting the robot to start an interaction, suspending or stopping an interaction session, etc. Many gestures of type 4 are consequently task independent.

The symbolic gestures tracking and recognition system for communicative gestures, based on the Mixed-State CONDENSATION, was enhanced through the definition of a new likelihood function. The fusion of shape and color cues was shown to significantly raise the recognition rate of static gestures. An enhanced Jump Markov Particle Filter is being implemented, which is expected to lead to better performances for dynamic gestures segmentation. These results are detailed in the PhD thesis manuscript of Ludovic Brèthes [8] and an excerpt can be found in [9].

The hand configuration recognition system will be implemented on the Rackham robot in KE2, in order to enable an elementary gesture-based interaction.

For preliminary testing and evaluation of the activity recognition system (CF-ACT) a set of 11 activities were selected and recorded. The list of activities consists of balance, bow, call, clap, flap, handshake, kick, manipulate, sit, walk, wave[4]. For each activity 10 example sequences were recorded

[4] Note, these activities were selected purely on terms of testing the system. In the future, further collaboration with UH will identify activities that actually occur in human-robot interaction.

for one male and one female subject. This gives in total 220 sequences with together 21222 frames. The recorded sequences were manually segmented and assigned its correct activity. The 220 segmented and classified sequences were then prepared into data sets suitable for training and classification. 50% of the recorded data were used for training, the other 50% for testing. To avoid *over fitting* in the NN classifier, the training data again has been divided into a training set and a validation set. The definition of features to be used for classification resulted in 120 intrinsic and 2 extrinsic features.

For evaluation, the main focus was laid on how the different features selection mechanisms influenced the recognition rates. The first evaluation started with an initial set of features and new features were added until the best recognition results were achieved in order to show the best overall possible recognition rates. The recognition of an activity was rated into four categories, *correct*, when the corresponding NN fired greater than 0.7, *ambiguous*, when at least one other NN fired greater than 0.7, *incorrect*, when one ore more NN fired greater than 0.7 but not the corresponding one and *missed*, when none of the NN fired greater than 0.7.

The second evaluation compared the recognition rates achieved with automated feature selection mechanisms with the rates from the first evaluation (hand-selected and optimal feature set). For MIFS a maximum of 5 and 10 features was chosen. The following table shows the overall recognition rates of the different feature selection methods.

Method	Correct	Ambiguous	Incorrect	Missed
Primitive	84.0 %	0.8 %	0.9 %	14.4 %
All features	98.3 %	0.6 %	0.0 %	1.1 %
MIFS 5 features	92.3 %	2.1 %	0.1 %	5.5 %
MIFS 10 features	95.7 %	1.6 %	0.0 %	2.7 %
HCFS 5 features	95.2 %	0.9 %	0.8 %	3.1 %

Table 1 Average recognition rates

A detailed description of the setup, the experiments and a discussion of the results can be found in [2].

The Object Attention System was implemented successfully on the demonstrator for the KE1 and integrated in its architecture. The achieved results [5, 7] are demonstrated along various examples of human robot interaction were ambiguities or missing input is resolved by the implemented system. By using pointing gestures even unknown objects can be addressed and consequently a view of the object can be used for learning the object. The implementation of the system has some restrictions so far, like the missing capability to locate objects in the global Cartesian map so that no autonomous navigation for finding objects is possible yet.

3.4 Future work

In close collaboration with RA3, several user studies reflecting situations of the KE's are planned for the next few months. These evaluations will lead to a suitable classification of activities and gestures.

For the communicative gesture recognition, efforts to final developments on dynamic gestures segmentation for 2D interpretation will be undertaken. The development and evaluation of an efficient Jump Markov Particle Filter will be finalized.

For the activity recognition, the main effort will be put on the improvement of the classification method. Usage of FFNNs has proved to be effective, but other classifiers may perform even better.

Classifiers with and without intrinsic time model will be implemented and evaluated, such as HMM, Bayesian Networks and SVM. Additionally, rule-based systems will be introduced to augment the performance of the classification. Extrinsic features will be incorporated. These have to be selected carefully, as there also has to be a recognition module for each of them. Obviously, special extrinsic features considerably simplify recognition of the associated activities.

The object attention system developed so far will be further evaluated and extended by the recognition of simple manipulative actions in a dialogue context. Hereby, the fusion of hand and object tracking with additional context knowledge will be investigated.

3.5 References

[1] S. Vacek, S. Knoop, R. Dillmann, ***Classifying Human Activities in Household Environments***, In: Modelling Others from Observation (MOO 2005), July 30, Edinburgh, Scotland, 2005

[2] S. Knoop, S. Vacek, R. Dillmann, S. Brännström, H. I. Christensen, ***Extraction, Evaluation, Selection and Classification of Motion Features for Human Activity Recognition***, Internal Report, Universität Karlsruhe, 2005

[3] Chrystopher L. Nehaniv, Kerstin Dautenhahn Jens Kubacki, Martin Haegele, Christopher Parlitz, Rachid Alami, ***A Methodological Approach relating the Classification of Gesture to Identification of Human Intent in the Context of Human-Robot Interaction***, *** IEEE *** International Workshop on Robot and Human Interactive Communication (IEEE RO-MAN 2005).

[4] Ludovic Brethes, Frédéric Lerasle, Patrick Danès, ***Data fusion for visual tracking dedicated to* human-robot interaction** , ICRA 2005

[5] A. Haasch, N. Hofemann, J .Fritsch, G. Sagerer, ***A Multi-Modal Object Attention System for a Mobile Robot***, Int. Conf. on Intelligent Robots and Systems (IROS) 2005

[6] B. Möller, S. Posch, A. Haasch, and G. Sagerer, ***Interactive Object Learning for Robot Companions using Mosaic Images***, Proceedings of the International Conference on Intelligent Robots and Systems, 2005

[7] Shuyin Li, Axel Haasch, Britta Wrede, Jannik Fritsch, Gerhard Sagerer. ***Human-style interaction with a robot for cooperative learning of scene objects.*** Int. Conf. on Multi-modal Interfaces (ICMI2005)

[8] Ludovic Brèthes. **Visual Tracking using Particle Filters. Application to H/R interaction**. PhD Thesis, LAAS-CNRS and Paul Sabatier University, December 2005.

[9] Excerpt from http://www.laas.fr/~lbrethes/KE2_2k5/

4 Appendix

Papers related to *Detection and perception of body parts based on sensor features (WP2.1):*

Z. Zivkovic, A.T. Cemgil, B. Krose, ***Approximate Bayesian methods for kernel-based object tracking***, journal paper under revision.

Ludovic Brethes, Frédéric Lerasle, Patrick Danès, ***Data fusion for visual tracking dedicated to human-robot interaction*** , ICRA 2005

Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann, ***Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP*** , International Conference on Humanoid Robots (Humanoids 2005), Tsukuba , Japan

Excerpt from http://www.laas.fr/~lbrethes/KE2_2k5/

Papers related to *Human body model: Integration and fusion (WP2.2)*

S. Knoop, S. Vacek and R. Dillmann, ***Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model***, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006), May 15, Orlando, Florida.

T. Spexard, J. Fritsch, M. Kleinehagenbrock, and G. Sagerer, ***Human-like Person Tracking with an Anthropomorphic Robot,*** Proceedings of the IEEE International Conference on Robotics and Automation, 2006, to appear.

Papers related to *Context based interpretation and classification of activities (WP 2.3)*

Chrystopher L. Nehaniv, Kerstin Dautenhahn Jens Kubacki, Martin Haegele, Christopher Parlitz, Rachid Alami, ***A Methodological Approach relating the Classification of Gesture to Identification of Human Intent in the Context of Human-Robot Interaction***, *** IEEE *** International Workshop on Robot and Human Interactive Communication (IEEE RO-MAN 2005).

S. Vacek, S. Knoop, R. Dillmann, ***Classifying Human Activities in Household Environments***, In: Modelling Others from Observation (MOO 2005), July 30, Edinburgh, Scotland, 2005

S. Knoop, S. Vacek, R. Dillmann, S. Brännström, H. I. Christensen, ***Extraction, Evaluation, Selection and Classification of Motion Features for Human Activity Recognition***, Internal Report, Universität Karlsruhe, 2005

A. Haasch, N. Hofemann, J .Fritsch, G. Sagerer, ***A Multi-Modal Object Attention System for a Mobile Robot***, Int. Conf. on Intelligent Robots and Systems (IROS) 2005

B. Möller, S. Posch, A. Haasch, and G. Sagerer, ***Interactive Object Learning for Robot Companions using Mosaic Images***, Proceedings of the International Conference on Intelligent Robots and Systems, 2005

Shuyin Li, Axel Haasch, Britta Wrede, Jannik Fritsch, Gerhard Sagerer. ***Human-style interaction with a robot for cooperative learning of scene objects.*** Int. Conf. on Multi-modal Interfaces (ICMI2005)

Approximate Bayesian methods for kernel-based object tracking

Zoran Zivkovic, Ali Taylan Cemgil, Ben Kröse

Intelligent and Autonomous Systems Group
University of Amsterdam, The Netherlands
email:{zivkovic,cemgil,krose}@science.uva.nl

Abstract

We present a probabilistic framework for real-time tracking of complex non-rigid objects. We approximate the shape of the tracked object by an ellipse and model its appearance by histogram based features derived from local image properties (such as color). First, we provide an observation model that, given an image, defines a likelihood function over the position and shape parameters of the tracked object. Second, we present an efficient procedure to search for modes of this likelihood function, based on a natural extension of the 'mean-shift' [24] method. This local search procedure retrieves quickly the most likely position and shape parameters in the vicinity of an initial configuration. We discuss how to integrate this efficient search into an approximate Bayesian filtering scheme in a theoretically sound way. Finally, we compare, in terms of solution quality and computation time, a number of suboptimal and optimal sequential Bayesian tracking schemes: including the Kalman filter, the mixture Kalman filter and other sequential importance sampling (particle filtering) techniques.

Keywords: object tracking, approximate Bayesian filtering

1. Introduction

In broad terms, Bayesian approaches to object tracking rely on two main underlying components: a transition (object motion) model, that describes kinematic constrains in the evolution of the objects state, and a observation model that defines the likelihood of the object configuration given the current measurement. In principle, once the model is decided upon, tracking boils down to a posterior inference problem that needs to be carried out sequentially using numerical (or analytical) integration techniques [2, 11]. However, in vision based tracking, defining a realistic yet practical observation model for non-rigid objects is a challenging problem.

Naturally, there have been numerous attempts in machine vision to provide solutions to this problem. From the range of existing tracking algorithms at one end the approach is to explicitly model the relation between the state of the object and the appearance of each pixel from the image region occupied by the tracked object. Such models can be specific for a certain object class; for example models tailored specifically for humans [15, 31] where the state usually contains the angles between the body parts; or more generic models to be learned from data [28, 25]. An alternative line of approach employs appearance models that are robust to deformations. For example the individual object pixels can be modelled independently by simply ignoring their spatial aspects. In this way, the derived statistics can be made invariant to arbitrary permutations of the pixels in the object region which provides some invariance to deformations. Such an example is the histogram-based representation used in [35, 13]. There is a number of other approaches that lie somewhere in between. For example [5, 19, 36, 20, 34] use

histogram based representations that include spatial information. The contour based algorithms [8, 21] also lie somewhere in between since they focus on detailed modelling but only of the outer contour shape of the tracked object in an image. The appearance within the image region occupied by the object is often neglected altogether or addressed using some histogram based representation [37, 16].

In this paper we follow [10] where the shape of the tracked object is approximated by an ellipse and the appearance within the ellipse is described by a histogram based model. The obvious advantage of such a model is its simplicity and general applicability. Another advantage, that made this observation model rather popular, is the existence of efficient local search schemes to find the image region with a histogram most similar to the histogram of the tracked object [18, 10, 9, 32, 38].

Given an accurate observation model and an efficient search procedure, the simplest tracking technique would be to use the "best" object configuration from the previous frame as the starting point to find the "best" object configuration in a new frame. This rather naive heuristic implicitly makes use of the general assumption that an object does not change its configuration drastically between consecutive frames and local search proceeds iteratively by small changes starting from the initial configuration. While such schemas work surprisingly well in certain scenarios, from a theoretical perspective it is unsatisfactory that the transition model is implicitly replaced by the dynamics of the inference algorithm in the configuration space.

The Bayesian filtering framework allows us to elegantly incorporate our knowledge about the object motion model while retaining computational advantages provided by fast local search. There are two issues here that are often raised and that will be addressed in this paper. The first issue is regarding the intractability of exact Bayesian filtering due to the complex observation model: it is not obvious how to choose among many possible approximate techniques. This paper analyzes and compares a range of approximate Bayesian tracking schemes. The second issue is: while the local search (if provided) seems to be useful it is not clear how to integrate it in a theoretically sound way into some approximate Bayesian filtering scheme.

In this paper, we analyze the following heuristic approaches: First, the local search is used to find the likely object configuration and the complex observation model is summarized by local approximation around this likely configuration. For example, the observation model can be approximated by a single Gaussian function centered at the most likely object configuration. Given this Laplace approximation, tracking can be achieved analytically by the Kalman filter [9]. However, when there are several likely object configurations, the likelihood function will be multimodal. In such cases it is useful to approximate the observation model by a Gaussian mixture [3, 4]. Consequently, tracking can be performed using the mixture Kalman filter [6]. Another approach is to use a sampling scheme to solve the Bayesian filtering problem. The bootstrap particle filter is an obvious candidate [21]. However, direct application of the bootstrap filter requires sampling from the transition model and does not make use of a fast search procedure [29]. In this paper we propose a novel particle filter that is making use of the efficient search procedure for constructing a proposal distribution. Viewing the approximation as a proposal is attractive because we obtain the computational advantages without compromising theoretical convergence properties [11] that other heuristic approach fail to have. We evaluate and analyze the mentioned tracking schemes on a large set of data.

The paper is organized as follows: In Section 2, we introduce the observation model. The similarity between a histogram from an elliptical image region with the histogram of the tracked object is formulated as a probability model. This presents an obvious generalization of similarity measures that have been used previously [9, 19], but nevertheless allows us to combine features derived from different image modalities. In Section 3, we present the general Bayesian tracking framework and discuss the related problems. In Section 4 present how the natural extension of the mean-shift procedure from [38] can be used to search for the most likely object configuration according to our observation model. The procedure efficiently

solves previous problems with sudden object scale and shape changes as is illustrated in Figure 1. In Section 4 we discuss how the search procedure can be used within a range of Bayesian tracking schemes. Experiments are given in Section 5 and in Section 6 we provide the conclusions and our final remarks.

2 Observation model

In this section we define a probability model that relates the state s_t of an object at time t with the video frame I_t observed at time t. Throughout the paper the index $t = 1 \dots T$ denotes the discrete time (frame) index. Occasionally, when the time index is not relevant we will omit it. We will denote the value at the i'th pixel by $I_t(x_i)$. Here, x_i is a 2 dimensional vector the coordinates of that denotes the pixel location.

2.1 Object shape

Suppose we are given an arbitrary shape $\mathcal{S}$ in an image, specified by a set of pixel locations x_i, i.e., $\mathcal{S} \equiv \{x_i : i\text{'th pixel belongs to the object}\}$. We approximate the shape of a non-rigid object in an image by its first and second order moments – an elliptical region we denote by $\mathcal{S}^e$. The original shape $\mathcal{S}$ may have been initially selected manually or detected using some other algorithm, for example background subtraction [33]. If there are $N_{\mathcal{S}}$ pixels that belong to the object of interest, we define

$$\theta \equiv \frac{1}{N_{\mathcal{S}}} \sum_{x_i \in \mathcal{S}} x_i \text{ and } V \equiv \frac{1}{N_{\mathcal{S}}} \sum_{x_i \in \mathcal{S}} (x_i - \theta)(x_i - \theta)^T. \tag{1}$$

Here, the first moment vector θ denotes the center of the object in the image I. The matrix of second moments V, that encodes scale and orientation, is symmetric and positive definite. Consequently, the θ and V describe an arbitrary elliptical region. There are various alternative ways to parametrize an ellipse. We use here, similar to [23], the following parametrization:

$$s \equiv [\theta^T, scale_x, scale_y, skew]^T \tag{2}$$

where $scale_x$ and $scale_y$ are the scaling and $skew$ is the skew transformation obtained from from V using the unique Cholesky factorization:

$$V = \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix}^T \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix} \tag{3}$$

These parameters describe the affine transformation that transforms a unit circle to the given elliptical region. Occasionally, by a slight abuse of notation we will refer to the state s as $s = (\theta, V)$ to explicitly highlight the dependence on θ and V. Similarly, we will refer to the elliptical shape defined by s by $\mathcal{S}^e(s)$.

2.2 Object appearance using histogram based features

The appearance of an object is described by a set of M scalar features $r_1, ..., r_M$ that are extracted from the local area of an image I defined by $\mathcal{S}^e(s)$. We view each r_m for $m = 1 \dots M$ as "bins" of a histogram. We define a quantization function $b : \mathbb{P} \rightarrow \{1 \dots M\}$, that associates with each observed pixel value a particular bin index m. The pixel values $I(x_i)$ are elements of the set $\mathbb{P}$, for example $\mathbb{P} = [0, 255]^3$ for RGB images.

The value r_m of the m-th bin is calculated from the elliptical image region $\mathcal{S}^e(s = (\theta, V))$ using:

$$r_m(I, s) \equiv |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s)} \mathcal{N}(x_i; \theta, V) \;\; \delta\left[b(I(x_i)) - m\right], \tag{4}$$

where δ is the Kronecker delta function. The kernel function $\mathcal{N}$ is chosen such that pixels in the middle of the object have higher weights than pixels at the borders of the objects. A natural choice is a Gaussian kernel defined by: $\mathcal{N}(x; \theta, V) = |2\pi V|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\theta)^T V^{-1}(x-\theta)\right)$. The prefactor $|V|^{\gamma/2}$ in (4) discounts for the fact that in practice we use only the N_s pixels from a finite neighborhood of the kernel center. We disregard samples further than 2.5-sigma and it is easy to show that one should use $\gamma \approx 0.1$ in this case.

The smooth kernel function, besides acting as a discount factor for (arguably less reliable) pixels near the borders, more importantly enables a fast gradient based search [9].

2.3 Probabilistic observation model

We introduce for each feature r_m a probability density function $p(r_m(I, s))$, the particular form to be defined later. We assume that the features r_m are independent. Furthermore, we assume that each feature r_m is uninformative if computed outside the region defined by s. The log-likelihood of s in an image I can be defined by:

$$\log \mathcal{L}(s) = \log p(I|s) \propto \sum_{m=1}^{M} \log p(r_m(I, s)). \tag{5}$$

This likelihood function for an image frame I_t is the observation model. The likelihood can be viewed as a generalization of many different histogram similarity measures that are used in literature. For example if $p(r_m(I, s))$ is chosen to be a Gaussian $\mathcal{N}(r_m(I, s); o_m, \sigma^2)$, the log-likelihood becomes the sum of squared distances as in [19]. The mean o_m and the standard deviation σ can be estimated from a set of test images of the object. The often used Bhattacharyya coefficient based model [29, 3]:

$$p(I|s) \propto \exp(\sum_{m=1}^{M} \sqrt{r_m(I, s)}\sqrt{o_m}/\sigma^2) \tag{6}$$

can also be seen as a particular choice.

2.4 Additional features

In our experiments, we have used videos from a static camera. Therefore, we can use the features from a simple background/foreground segmentation scheme similar to [33]. We view the result of the background/foreground segmentation as an additional and independent observed image $\tilde{I}$ where $\tilde{I}(x_i) \in \{0, 1\} = \tilde{\mathbb{P}}$. We define a new quantization function $\tilde{b} : \tilde{\mathbb{P}} \to \{1, 2\}$ where $\tilde{m} = 1, 2$ denotes, say, background and foreground. We define a new set of features $\tilde{r}_{\tilde{m}}(\tilde{I}, s)$ as in (4). Similarly, we define $p(\tilde{I}|s)$ by defining $\tilde{o}$ and $\tilde{\sigma}$. Intuitively, this latter feature measures the ratio of background pixels to the foreground pixels in the elliptic region. Due to independent observation assumption, the contributions to the likelihood function will be additive, i.e.,

$$\log \mathcal{L}(s) = \log p(I|s) + \log p(\tilde{I}|s)$$

Clearly, the set of features could be extended further by more elaborate image processing methods. Possible choices include normalized color values, optical flow results [1], etc. Choosing the particular type of local image property, i.e., feature selection, is not the main focus of this paper as this highly depends upon the situation [7].

3 Inference

The observation model should be combined with our knowledge about the object motion. To simplify the notation, we will denote the observations as $z_t \equiv I_t$. If we use different image modalities as described above we have $z_t \equiv \{I_t, \tilde{I}_t\}$. To track an object, we wish to estimate the density $p(s_t|z_{1:t})$ for each $t = 1, 2 \ldots$ given the sequence of measurements $z_{1:t}$. This density is also known as the *the filtering density*. In a Bayesian setting, the filtering density is assumed to represent our entire knowledge about the current state s_t of the tracked object and all desired quantities (such as most likely position) can be derived from this density. An important observation is that $p(s_t|z_{1:t})$ can be calculated recursively and online if the dynamic model is described by a first order Markov process $p(s_t|s_{t-1}) = p(s_t|s_{1:t-1})$ and the measurements are independent from each other given the latent dynamic process, i.e., $p(z_t|s_t)p(z_t|s_{1:T})$. Recursive updates can be performed using the prediction stage

$$p(s_t|z_{1:t-1}) = \int p(s_t|s_{t-1})p(s_{t-1}|z_{1:t-1})ds_{t-1} \tag{7}$$

and the update stage

$$p(s_t|z_{1:t}) = \frac{1}{c}p(z_t|s_t)p(s_t|z_{1:t-1}) \tag{8}$$

where $c = \int p(z_t|s_t)p(s_t|z_{1:t-1})ds_t$ is a normalization constant and $p(s_{t-1}|z_{1:t-1})$ is the previous estimate.

We use a simple linear dynamic model $p(s_t|s_{t-1}) = \mathcal{N}(s_t; As_{t-1}, Q)$. We further assume that the transition matrix $A = I$ is an identity matrix and transition noise covariance Q to be diagonal, with values estimated from data. Clearly, more elaborate dynamical models than this simple random walk model can be estimated from training data, e.g., see [21]. The issues related to non-linear motion models are discussed for example in [22].

The complex form of the observation function (5) renders the update step (8) analytically intractable. Complex observation models often occur in machine vision applications. In Section 5, we will investigate two approximation strategies to circumvent this. The approximations rely on the local search method described in the next section.

4 Searching the modes of the observation model

We propose here an efficient and specialized gradient descent procedure to search for the likely object configurations. The local search starts with some starting point $s^{\{k\}}$. Here the superscript $\{k\}$ denotes the iteration index. Similar to [10, 19, 36, 20, 34] the gradient descent step is calculated using two phases: first the similarity measure (5) is approximated locally using a Taylor expansion, then the gradient step is calculated. Instead of the the mean-shift procedure that gives a gradient step only for the object position θ [9, 18, 32] we use an extended version of the mean-shift to get the gradient step for all the parameters of the ellipse from s.

4.1 Approximating log-likelihood

We approximate the observation model (5) by first order Taylor expansion (in r_m around $r_m(I, s^{\{k\}})$):

$$\log \mathcal{L}(s) \approx c + \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} r_m(I, s) \tag{9}$$

where c is a constant term. We denote the variable term from above by $f(s)$ and replace (4):

$$f(s) = \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \mathcal{N}(x_i; \theta, V) \delta\left[b(I(x_i)) - m\right] = |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(x_i; \theta, V), \tag{10}$$

where

$$\omega_i = \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} \delta\left[b(I(x_i)) - m\right]. \tag{11}$$

For the Bhattacharyya coefficient metric we have:

$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{r_m(I, s^{\{k\}})}} \delta\left[b(I(x_i)) - m\right]. \tag{12}$$

4.2 Mean-shift extension using Jensen's lower bound

The previous approximation leads to a functional form (10) that resembles a kernel based density estimate. The mean-shift algorithm [24] can be used to calculate the gradient step on (10) with respect to the object position θ as in [10]. As described in [26] the mean-shift step is derived from a lower bounding function of (10). The lower bound follows from the convexity of the kernel function. From the Jensen's inequality, typical for variational approaches [27], we can get a different lower bound:

$$\log f(s) \geq G(s, q_1, ..., q_N) = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \log\left(\frac{\omega_i |V|^{\gamma/2} \mathcal{N}(x_i; \theta, V)}{q_i}\right)^{q_i} \text{ where } \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i = 1 \text{ and } q_i \geq 0. \tag{13}$$

It is easy to show that for a given s the equality sign in (13) is achieved for:

$$q_i = \frac{\omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}})}{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(\vec{x}_i; \theta^{\{k\}}, V^{\{k\}})}. \tag{14}$$

Given the q_i-s we maximize the part of G that depends on the parameters:

$$g(s) = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i \log(|V|^{\gamma/2} \mathcal{N}(x_i; \theta, V)). \tag{15}$$

For the Gaussian kernel and $\frac{\partial}{\partial \theta} g(s) = 0$ we get:

$$\theta^{\{k+1\}} = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i x_i = \frac{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \vec{x}_i \omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}})}{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}}).} \tag{16}$$

Note that this update equation for the position estimate is equivalent to the mean-shift update for the Gaussian kernels. An advantage is that we can now derive simple equations for updating V. For Gaussian kernel from $\frac{\partial}{\partial V} g(\theta, V) = 0$ we get:

$$\vec{V}^{\{k+1\}} = \beta \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i (x_i - \theta^{\{k\}})(x_i - \theta^{\{k\}})^T \tag{17}$$

where $\beta = 1/(1 - \gamma)$.

a) the selected region b) mean-shift c) extended mean-shift

Figure 1: Searching for the most similar image region - the mean shift and the extended algorithm.

4.3 Practical iterative algorithm

For the sake of clarity we present here the whole algorithm:

Input: starting shape $s^{\{k\}}$ $(k = 0)$, the new image I.

1. For $s^{\{k\}}$ calculate the r_m-s (4) and the weights (11).
2. Calculate q_i-s using (14) and the new estimates $\theta^{\{k+1\}}$,$V^{\{k+1\}}$ using (16-17).
3. Line search if needed.
4. If no new pixels are included using the new elliptical region defined by the new estimates $\theta^{\{k+1\}}$ and $V^{\{k+1\}}$ stop, otherwise set $k \leftarrow k + 1$ and go to 1.

Because of the additional local approximation (9) we need an additional a line search step to get a proper gradient descent procedure [14]. For many distance measures the second condition from (13) is not satisfied since the weights ω_i from (11) are not always nonnegative. In such cases the update steps are still in the gradient direction but the line search is advisable. For the Bhattacharayya coefficient measure the weights are always positive and in practice it turns out that line search is not necessary since (9) appears to be often a good local approximation as in [10, 19, 36, 20, 34].

In Figure 1 we illustrate the performance of the algorithm. A player is selected as indicated by the ellipse in Figure 1a. The image is scaled 1.5 times in the vertical direction and then rotated 45 degrees as presented in Figure 1b. We use the initial shape of the region and we manually select a nearby random position in the new rotated and scaled image. The iterations and the final of the mean-shift procedure [10] are presented in Figure 1b. In Figure 1c we present the iterations of our algorithm. Both the shape and the position are accurately estimated.

5 Approximate Bayesian tracking schemes

We present here a range of approximate Bayesian tracking schemes and show how to make use of the efficient local search scheme from the previous section.

5.1 Laplace approximation and the Kalman filter

We approximate the observation model $p(z_t|s_t)$ by Laplace approximation. This replaces the exact observation model with a Gaussian distribution $\mathcal{N}(s_t; m(z_t), R(z_t))$. We denote the mean $m(z_t)$ and

covariance $R(z_t)$ to highlight the fact that these parameters are calculated after observing z_t.

Consequently, the filtering density can be represented by a Gaussian $p(s_t|z_{1:t}) \approx \mathcal{N}(s_t; \mu_t, P_t)$ and the equations (7) and (8) become the well known Kalman filtering equations (specialized for the case when the observation matrix $C = I$):

$$\begin{aligned} p(s_t|z_{1:t-1}) &\approx \mathcal{N}(s_t; \mu_{t|t-1}, P_{t|t-1}) = \mathcal{N}(s_t; A\mu_{t-1}, A^T P_{t-1} A + Q) && (18)\\ p(s_t|z_{1:t}) &\approx \mathcal{N}(s_t; \mu_t, P_t) && (19)\\ K_{\text{gain}} &= P_{t|t-1}(P_{t|t-1} + R(z_t))^{-1} && (20)\\ \mu_t &= \mu_{t|t-1} + K_{\text{gain}}(m(z_t) - \mu_{t|t-1}) && (21)\\ P_t &= (I - K_{\text{gain}})P_{t|t-1} && (22) \end{aligned}$$

At each time slice, before the update step, we calculate the mean $m(z_t)$ using the search procedure introduced in Section 4 and use the local curvature to calculate $R(z_t)$. The search is started from the predicted mean $\mu_{t|t-1}$.

Whilst simple to implement, there are two problems with this approach. First, representing $p(z_t|s_t)$ by a single Gaussian works well when the object is clearly visible but not in case of occlusions and background clutter. Secondly, since we use just the $\mu_{t|t-1}$ as the starting point for the local search procedure, finding the dominant mode after the occlusions or sudden movements is difficult. In such cases, the tracking will usually fail as illustrated in Figure 3a.

5.2 Mixture Kalman filter

A more accurate approximation to the observation model can be obtained by using a Gaussian mixture rather than a single Gaussian. Such an approach is more likely to capture the multimodal nature of $p(z_t|s_t)$. Unfortunately, optimally fitting a mixture distribution to a target model (e.g. in terms of KL divergence) is intractable. Fortunately, there exist powerful heuristics that seem to result in quite satisfactory performance.

The key to a good approximation is capturing the various modes of the multimodal $p(z_t|s_t)$. To locate different modes, we use the deterministic local search procedure introduced in Section 4, initialized from K different start positions. A natural choice for generating the K starting points is to sample from the predictive distribution $p(s_t|z_{1:t-1})$. Yet, a more robust way is to sample from a mixture [3]

$$\alpha p(s_t|z_{1:t-1}) + (1 - \alpha)u(s_t). \tag{23}$$

Here α is a mixing parameter that allows samples both from the prediction $p(s_t|z_{1:t-1})$ and from some wide distribution $u(s_t)$ (for example uniform over the whole image). By tuning α, one can adjust the amount of "surprise", as well as discount for the fact that we carry forth only an approximation of the exact predictive distribution – we have used $\alpha = 0.9$ in our experiments. The result of the local search will be $K(z_t) \leq K$ modes of $p(z_t|s_t)$ (as in [3] we detect when two searches end up in the same mode).

We could use only the most dominant mode and use the Kalman filter as introduced in 5.1. On the other hand, especially in case of occlusion, the posterior has a number of pronounced modes. Typically, each of these modes may be associated with a different hypothesis about the evolution of the state s. The correct trajectory in the state space can only be disambiguated after observing the future data and discarding a mode may cause the tracker to miss the track. Therefore, we decide to keep all the $K(z_t)$ modes and to approximate the complex $p(z_t|s_t)$ by a Gaussian mixture:

$$p(z_t|x_t) \approx \sum_{j=1}^{K(z_t)} \rho(z_t)^j \mathcal{N}(x_t; m(z_t)^j, R(z_t)^j) \tag{24}$$

where the superscript $j = 1 \dots K(z_t)$ denotes the components of the mixture. Here, $\rho(z_t)^j$ denotes the weight of the j'th mixture component.

Suppose the filtering density at time $t-1$ be approximated by a mixture with $i = 1 \dots K$ components:

$$p(s_{t-1}|z_{1:t-1}) \approx \sum_{i=1}^{K} w_{t-1}^i \mathcal{N}(s_{t-1}; \mu_{t-1}^i, P_{t-1}^i)$$

The prediction is computed for each mixture component using:

$$\begin{aligned} p(s_t|z_{1:t-1}) &\approx \sum_{i=1}^{K} w_{t-1}^i \mathcal{N}(s_t; \mu_{t|t-1}^i, P_{t|t-1}^i) \\ &= \sum_{i=1}^{K} w_{t-1}^i \mathcal{N}(s_t; A\mu_{t-1}^i, A^T P_{t-1}^i A + Q) \end{aligned}$$

The updated density is given by

$$\begin{aligned} p(s_t|z_{1:t}) &\approx \sum_{i=1}^{K} \sum_{j=1}^{K(z_t)} w_t^{ij} \mathcal{N}(s_t; \mu_t^{ij}, P_t^{ij}) \\ K_{gain}^{ij} &= P_{t|t-1}^i (P_{t|t-1}^i + R(z_t)^j)^{-1} \\ \mu_t^{ij} &= \mu_{t|t-1}^i + K_{gain}^{ij}(m(z_t)^j - \mu_{t|t-1}^i) \\ P_t^{ij} &= (I - K_{gain}^{ij}) P_{t|t-1}^i \\ \pi_t^{ij} &= \mathcal{N}(m(z_t)^j; \mu_{t|t-1}^i, P_{t|t-1}^i + R(z_t)^j) \\ w_t^{ij} &= w_{t-1}^i \rho(z_t)^j \pi_t^{ij} \end{aligned}$$

Note that by starting with a mixture of K components, we have ended up with a mixture with $K \times K(z_t)$ components. Obviously, such an update would lead to exponentially growing number of components with increasing t. There are various schemes to approximate this new mixture by a mixture with lower number of components. A natural one is to view w_t^{ij}-s as a multinomial distribution on $(1 \dots K) \times (1 \dots K(z_t))$ in the double index (ij), to sample K samples from this distribution and retain the selected modes as in [6].

The performance using only $K = 5$ modes is illustrated in Figure 3b. The algorithm was able to approximate the multimodal nature of $p(z_t|s_t)$ in case of occlusion and to recover very fast after the occlusion.

5.3 Sampling and Sequential Monte Carlo

In sampling based methods, we wish to approximate a target density (e.g. a posterior distribution) by a discrete uniform distribution defined on a set of points. Ideally, we wish to generate a set of points (samples, particles) $\{s^{(i)}, i = 1 \dots N\}$ such that

$$p(s) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(s - s^{(i)}) \tag{25}$$

where δ is the Dirac delta function. This approximation is in the sense that expectations of arbitrary functions $f(s)$ under p are approximated by averages as

$$\int f(s)p(s)ds \approx \int f(s)\frac{1}{N}\sum_{i=1}^{N}\delta(s-s^{(i)})ds = \frac{1}{N}\sum_{i=1}^{N}f(s^{(i)})$$

Sequential Monte Carlo (SMC) [11], a.k.a. particle filtering, is a powerful technique for generating samples from a target posterior distribution. SMC is especially useful in tracking scenarios, where observations arrive sequentially.

Suppose we have a sample based approximation to the filtering density at time $t-1$ (see Section 3) as

$$p(s_{t-1}|z_{1:t-1}) \approx \sum_{i=1}^{N}\tilde{w}_{t-1}^{(i)}\delta(s_{t-1}-s_{t-1}^{(i)})$$

This equation is similar to (25) but we have introduced *normalized weights* $\tilde{w}^{(i)} \geq 0$ such that $\sum_i \tilde{w}^{(i)} = 1$. Hence, the filtering distribution is represented by a set of particles and their associated weights $\{\tilde{w}_{t-1}^{(i)}, s_{t-1}^{(i)}, i = 1 \dots N\}$. The essential idea of SMC is to evolve this representation into a new set of weights and particles $\{\tilde{w}_t^{(i)}, s_t^{(i)}, i = 1 \dots N\}$ via (7) and (8) when the observation z_t becomes available at time t. The common practice is to use importance sampling to resolve the following basic issues: (1) How to generate a new set of samples, and (2) How to compute the new weights.

5.3.1 Importance Sampling

Suppose we are somehow given a distribution $q(s)$ that is easy to sample from (e.g. a Gaussian). We call the q distribution a *proposal distribution*. We generate N independent samples $s^{(i)}$ from this proposal and obtain a sample based approximation as $q(s) \approx \sum_{i=1}^{N}\delta(s-s^{(i)})/N$. Then we can approximate

$$p(s) = \frac{p(s)}{q(s)}q(s) \approx \sum_{i=1}^{N}\tilde{w}^{(i)}\delta(s-s^{(i)}) \tag{26}$$

where $\tilde{w}^{(i)} = w^{(i)}/\sum_{j=1}^{N}w^{(j)}$ and $w^{(i)} = p(s^{(i)})/q(s^{(i)})$. One can interpret the $\tilde{w}^{(i)}$ as correction factors to compensate for the fact that we have sampled from the "incorrect" distribution $q(s)$. An important feature of importance sampling is that it (in principle) works even when the exact expression of $p(s)$ is not know. All we need is a function $p^*(s)$, that we can evaluate pointwise which is proportional to p, i.e., $p(s)(1/Z)p^*(s)$, and the normalized weights would be the same. This is important in Bayesian inference schemes since it often occurs that it is difficult to calculate the normalizing constant Z.

5.3.2 Sequential Importance Sampling

Now we apply importance sampling to compute the filtering distribution $p(s_t|z_{1:t})$ for tracking. We start from the joint posterior distribution over all time slices

$$p(s_{1:t}|z_{1:t}) = \frac{1}{Z_t}\prod_{k=1}^{t}p(z_k|s_k)p(s_k|s_{k-1}) \equiv \frac{1}{Z_t}p^*(s_{1:t}|z_{1:t}) \tag{27}$$

The key idea in *sequential importance sampling* is the sequential construction of the proposal distribution, possibly using the available observations $z_{1:t}$, i.e.,

$$q(s_{1:t}|z_{1:t}) = \prod_{k=1}^{T} q(s_k|s_{k-1}, z_{1:k})$$

Given a sequentially constructed proposal distribution, one can compute the importance weight recursively as follows:

$$w_t^{(i)} = \frac{p^*(s_{1:t}^{(i)}|z_{1:t})}{q(s_{1:t}^{(i)}|z_{1:t})} = \frac{p(z_t|s_t^{(i)})p(s_t^{(i)}|s_{t-1}^{(i)})}{q(s_t^{(i)}|s_{t-1}^{(i)}, z_{1:t})} \underbrace{\frac{p^*(s_{1:t-1}^{(i)}|z_{1:t-1})}{q(s_{1:t-1}^{(i)}|z_{1:t-1})}}_{=w_{t-1}^{(i)}} \tag{28}$$

Note that in this notation $s_{1:t}^{(i)}$ denotes a full trajectory. The desired particles that represent the filtering distribution $p(s_t|z_{1:t})$ are obtained simply by throwing away $s_{1:t-1}^{(i)}$, i.e., by keeping the samples at the time slice t and weights $w_t^{(i)}$. In practice, we don't need to store the full trajectory, because we have chosen a proposal that depends only upon $s_{t-1}^{(i)}$.

The sequential update schema is potentially more accurate than naive importance sampling with a fixed proposal. This is because at each step t, one can construct a fairly accurate proposal distribution that takes the current observation z_t into account. Clearly, many different proposal distributions can be envisaged. A simple and popular choice for the proposal is to select the transition model. Here, we have the proposal

$$q(s_t|s_{t-1}^{(i)}, z_{1:t}) = p(s_t|s_{t-1}^{(i)}) \tag{29}$$

Note that this proposal is independent of the current observation z_t. Substituting this into (28) and simplifying, we see that

$$w_t^{(i)} = p(z_t|s_t^{(i)})w_{t-1}^{(i)}$$

This is known as likelihood weighting or the bootstrap filter [17]. A more natural and accurate choice for the proposal distribution would have been the filtering distribution given as

$$q(s_t|s_{t-1}^{(i)}, z_{1:t}) = p(s_t|s_{t-1}^{(i)}, z_{1:t}) \tag{30}$$

In this case the weight update rule in Eq. 28 simplifies to

$$w_t^{(i)} = p(z_t|s_{t-1}^{(i)})w_{t-1}^{(i)}$$

In fact, provided that the proposal distribution q is constructed sequentially and past sampled trajectories are not updated, the filtering distribution is the optimal choice in the sense of minimizing the variance of importance weights $w^{(i)}$ [12].

Unfortunately, exact calculation of this optimal proposal is also intractable due to the complex observation model. Therefore, we will use a proposal distribution calculated by the mean-shift procedure, as introduced in previous section.

5.3.3 Selection

In practice, the sequential importance sampling may be degenerate. After a few iterations of the algorithm, only one particle has almost all of the probability mass and most of the computation time is wasted for updating particles with negligible probability. In fact, it can be shown (e.g. [12]) that the variance of $w_t^{(i)}$ indeed increases unboundedly with t.

To avoid the undesired degeneracy problem, several heuristic approaches are proposed in the literature. The basic idea is to duplicate or discard particles according to their normalized importance weights. The selection procedure can be deterministic or stochastic. Deterministic selection is usually greedy; one chooses N particles with the highest importance weights. In the stochastic case, called *resampling*, particles are drawn with a probability proportional to their importance weight $w_t^{(i)}$. Recall that normalized weights $\{\tilde{w}_t^{(i)}, i = 1 \dots N\}$ can be interpreted as a discrete distribution on particle labels (i).

A summary of the particle filtering algorithm is as follows. Before we start, we need to choose N, number of particles, and the form of the proposal distributions $q(s_t|y_t, s_{t-1})$ for $t = 1 \dots T$ and $q(s_0)$. We assume $p(s_1|s_0) \equiv p(s_1)$ and start with N initial samples $s_0^{(i)}$ from some $q(s_0)$ and with initial weights $w_0^{(i)} = 1$. For each new t we repeat:

1. Generate new samples $s_t^{(i)}$ from $q(s_t|z_t, s_{t-1}^{(i)})$.
2. Calculate weights $w_t^{(i)}$ using (28) and the normalized weights $\tilde{w}_t^{(i)}$.
3. (Optional resampling:) Randomly select N samples $s_{new}^{(j)}$ from $s_t^{(i)}$. Each sample $s_t^{(i)}$ is selected with probability equal to its normalized weight. The new samples are used further $s_t^{(i)} \leftarrow s_{new}^{(j)}$ with weights $w_t^{(i)} = 1$.

In our experiments we used the simple proposal (29), (the bootstrap filter) and the proposal computed using the mean-shift procedure. The performance is illustrated in Figure 3.

6 Experiments

First we present some experiments to compare the new local search scheme from section 4 to previously proposed schemes. Then we evaluate the various approximate Bayesian tracking schemes from the previous section.

6.1 Extended mean shift and the mean-shift tracking

The 'hand' sequence is used to demonstrate the full 5-DOF color-histogram-based tracking. To be robust to light conditions we used again 8×8 histogram in the hue-saturation color space. The hand is tracked. The sequence has 256 frames and the position and the shape of the hand are changing rapidly. In Figure 2c we can see that the new algorithm can track the hand and also adapt to the shape of the object. We simply used the "best" position from the previous frame to start the search for new image.

Finally, in Figure 2d we present the number of iterations of the algorithm for the 'hand' sequence. The computational complexity of one iteration of the new algorithm is only slightly higher than the computational complexity of the mean-shift. The average number of iterations per frame was approximately 6. This is slightly more then 4 (4.6 in our experiments for our sequence) that was reported for the mean-shift

based iterations in [10]. This amount should be multiplied by 3 if the simple scale adaptation is used [9] where the algorithm is additionally tested with a 10% larger and a 10% smaller ellipse. Empirically we measured average of 15.1 iterations in our experiments for our sequence.

6.2 Comparing different Bayesian tracking schemes

We selected a number of sequences from the public available annotated data set (http: //homepages.inf.ed.ac.uk /rbf /CAVIAR). Our data set contains 7 surveillance videos of 3 different scenes and with different scene viewing angles. We used different tracking schemes to track the 15 different people from these sequences. The ground truth bounding boxes for the tracked people were used to evaluate the tracking algorithms. The frame from a video where a person appears and the corresponding ground truth bounding box were used to initialize the object model (histogram-based appearance and elliptical shape). The person is then tracked until it leaves the field of view. In total 3365 frames were used for evaluation.

We use the following relative overlap measure to evaluate the performance of various algorithms. Let R_{gt} be the image region defined by the ground truth bounding box of the tracked object for a given frame. Let R_e be the elliptical region estimated by an algorithm from the algorithms we described. The relative overlap is defined by:

$$overlap = \frac{R_e \cap R_{gt}}{R_e \cup R_{gt}} \tag{31}$$

where $R_e \cap R_{gt}$ is the intersection and $R_e \cup R_{gt}$ is the union of the two image regions. The overlap is computed numerically and has values between 0 and 1. The performance of different tracking schemes is illustrated for a short sequence in Figure 3.

In Table 6.2 we present the results of the evaluation for the whole data set. Note that since we compare the ground truth bounding box with the estimated elliptical region the overlap will be on average smaller that actual. Overlap of 0.4 of two ellipses visually still looks reasonable. We report the percentage of total number of frames where $overlap > 0.4$,$overlap > 0.3$ and $overlap > 0.2$.

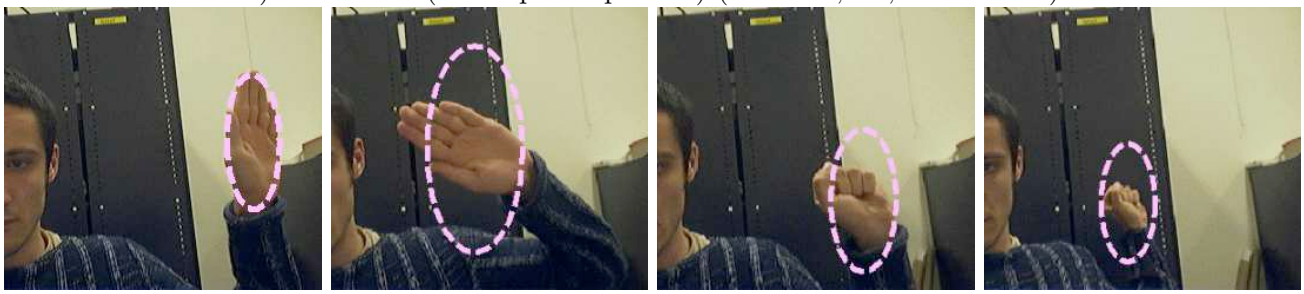
For the realistic data there are many situations when there are many regions in the image with similar colors histogram as the object. All the schemes perform poor when only color histogram features were used. When we extend the representation with the foreground/background segmentation histogram based features (see Section 2.4) many local maxima of the likelihood are suppressed and the tracking greatly improves. Here the Kalman filter performs the worst since it relies on the roughest approximation of the likelihood function. Occlusions are a common problem in tracking. In the last part of the table we report the results for the more challenging situation where we add larger occlusions to the sequences by adding a white stripe over the images. The Kalman filter performs poorly. The simple bootstrap particle filter has also difficulties in handling occlusions and sudden movements. On the other hand the performances of the mixture Kalman filter and the particle filter with proposal distribution are only slightly decreased.

6.3 Comparing computational complexity

In order to compare the mentioned tracking schemes we provide here a rough estimate of their computational complexity. We focus on the major part of the computation, namely the costs of approximating the observation model. The costs will be presented relative to the mean-shift based Kalman filter form [9].

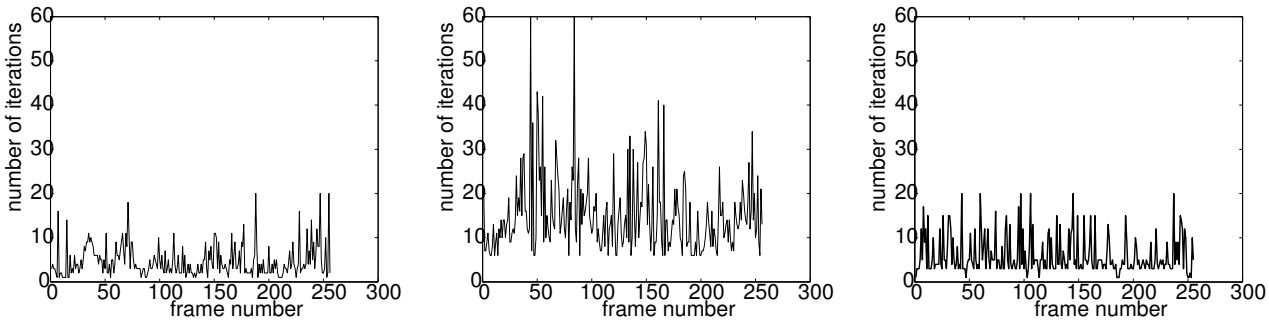
The Kalman filter and the mixture Kalman filter use the local search for approximating the observation model. Let $N_{iteration} \approx 6$ be the average number iterations. In each iteration the histogram of the current

a) Mean-shift (no shape adaptation) (frames 0,100,200 and 250)



b) Mean-shift (simple shape adaptation)(frames 0,100,200 and 250)

c) Extended mean-shift (frames 0,100,200 and 250)



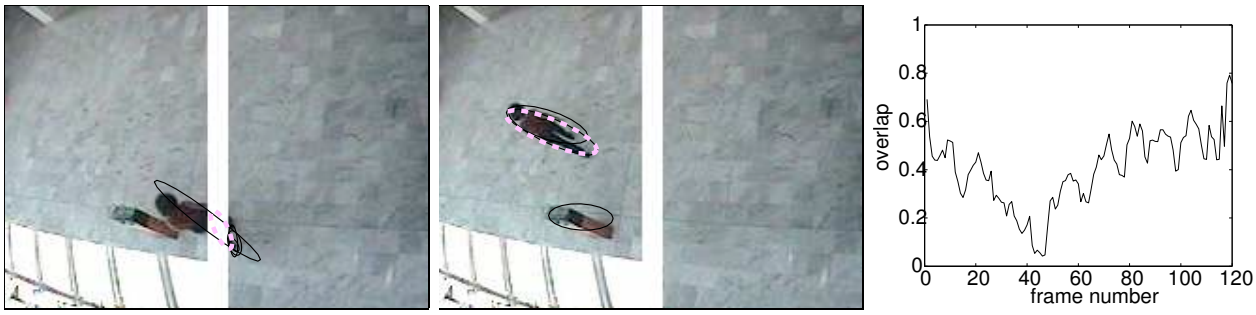
Mean-shift Mean-shift+shape Extended mean-shift

d) number of iterations per frame. The average number is 4.6 for the mean-shift with no shape adaptation, 15.1 for mean-shift with simple shape adaptation as in [9] and 6.3 for the new extended algorithm.

Figure 2: Illustrating the performance of the new algorithm as compared to the mean-shift. The estimated position and shape of the tracked objects is represented by the dashed ellipse.

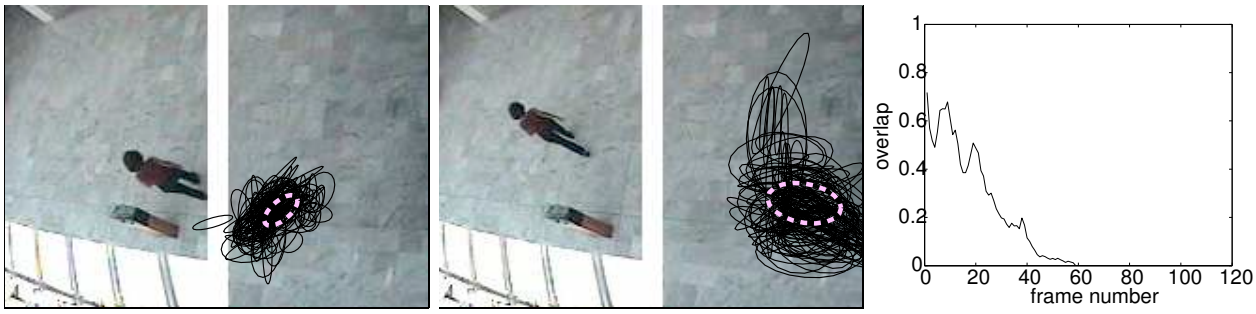
frame 42 frame 100 overlap

a) Kalman filter.



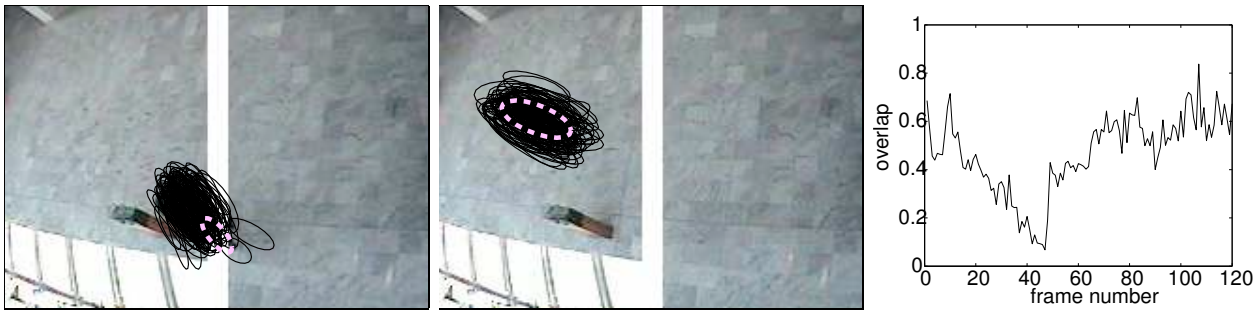
frame 48 frame 100 overlap

b) Mixture Kalman filter. The black ellipses present the modes of the Gaussian mixture.



frame 70 frame 100 overlap

c) Simple bootstrap particle filter with 100 particles. The black ellipses present the particles.



frame 48 frame 100 overlap

c) Particle filter with 100 particles and the proposal distribution. The black ellipses present the particles.

Figure 3: Illustrating how the various tracking schemes handle occlusion. We add the white stripe over the image to introduce the occlusion. The person was walking from right to the left side of the image. The maximum of the estimated density represented by the dashed ellipse is used as the estimated position and shape. Relative overlap with the ground truth bounding box is presented on the right.

Histogram type, occlusion	Relative overlap with the ground truth	Kalman filter	Mixture Kalman Filter (5 modes)	Bootstrap particle filter (100 particles)	Particle filter with proposal (100 particles)
color, no occlusion	average	0.33	0.34	0.31	0.32
	overlap>0.2	66.0%	67.0%	61.7%	59.3%
	overlap>0.3	55.9%	60.1%	53.9%	55.6%
	overlap>0.4	45.0%	46.7%	41.5%	42.0%
color+ background, no occlusion	average	0.44	0.48	0.50	0.50
	overlap>0.2	89.1%	91.8%	93.8%	93.7%
	overlap>0.3	74.1%	83.0%	83.9	84.3%
	overlap>0.4	55.9%	63.6%	66.8	67.9%
color+ background, with occlusion	average	0.35	0.45	0.40	0.46
	overlap>0.2	69.6%	89.5%	80.0%	92.6%
	overlap>0.3	64.0%	79.9%	67.9%	79.9%
	overlap>0.4	49.3%	60.9%	50.0%	61.0%

Table 1: Evaluation results on a data set of 7 videos, 3 scenes, 15 people, 3365 frames in total. Relative overlap with the ground truth bounding box is presented. The color histogram and the combination with the background subtraction were used. We also tried the sequences with additional occlusion where we placed a 20-pixel-wide white stripe over the images.

region is calculated. This involves putting each of the N_{points} from the region to its bin. Let c_H be the costs of reading a pixel value and placing it into its histogram bin. We further need to compute 2 weighted sums in (8) and 3 for V (V is symmetric) so we use $N_{sums} = 5$. Let c_S be the cost per element in each weighted sum. For the mixture Kalman filter the search is performed from K starting points and the total costs are:

$$KN_{iteration}N_{points}(c_H + N_{sums}c_S) \approx KN_{iteration}N_{points}(1 + N_{sums})c_S \tag{32}$$

It is difficult to compare c_H and c_S so for simplicity and without any proof we assumed that they are equal.

The mean-shift needs on average $N_{iteration} \approx 4$ and only two weighted sums N_{sums} which is 2.5 faster than the new extended version but the new procedure can handle the scale changes. The mean-shift with the simple scale adaptation involves running the procedure 3 times for different scales [9] which is slower than the new procedure and still does not solve the scale changes problems as demonstrated. We will report here the results relative to the mean-shift search [9]. In [9] they argue that the mean-shift search without scale adaptation is approximately 24 times faster than the exhaustive search within a region [13] under some realistic assumptions as reported in [9] (for rectangular regions the efficiency of the extensive search can be increased [30]).

The particle filter computation time scales with the number of particles $K_{particles}$. For each particle we need to compute the histogram. For comparing the histograms we need to compute the weighted sum (6). For simplicity we will assume the same c_S costs per element of this sum. If the number of bins m is close to the number of points within the region N_{points} we can approximate the computation costs by:

$$N_{particles}(N_{points}c_H + mc_S) \approx N_{particles}N_{points}2c_S \tag{33}$$

Relative with respect to the mean shift Kalman filter without scale adaptation	Kalman filter	Kalman filter simple scale adaptation [9]	Mixture Kalman Filter (5 modes)	Bootstrap particle filter (100 samples)	Particle filter (proposal distribution) (100 samples)
theoretical	2.5	3	12.5	16.7	29.2
empirical (m=64)	2.1	2.9	7.2	8.9	15.5
empirical (m=512)	2.1	2.9	5.2	9.0	15.6

Table 2: Approximate theoretical and empirical comparison of the computational times relative to the simple mean-shift based Kalman filter with no scale adaptation. The empirical results are reported for RGB color histogram appearance representation with $m = 64$ and $m = 512$ bins.

For the particle filter with the Gaussian mixture proposal we add (33) and the computational costs of the Gaussian mixture approximation (32).

In Table 6.3 we summarize the approximate costs from above. These approximations completely disregarded the additional overhead of the filtering schemes. We also report the empirical results that include these costs. We compared the processing time of the tracking schemes relative to the mean-shift based Kalman filter without scale adaptation. The reported empirical comparison will depend on implementation. We observe that the empirical results roughly follow the approximate calculations. Note that for histograms with small number of bins m the particle filters become more efficient relative to the Kalman filter and mixture Kalman filter as predicted by (33). All of the algorithms as suitable for real-time applications. For example the Kalman filter needs only a few milliseconds per frame on a 2GHz Pentium computer.

7 Conclusions

We considered a generally applicable observation model where the shape of the tracked object is approximated by an ellipse in general position and its appearance by histogram based features. We provide an efficient local search procedure to find the likely object configurations according to this observation model. Two issues are raised that are common for many visual tracking problems. First, due to the complexity of the observation model, the model does not admit an exact analytical treatment of the tracking problem so we have to resort to some approximate Bayesian filtering scheme. Second, it might be useful to integrate the efficient local search procedure into the filtering scheme.

We analyzed a range of heuristic approaches. First, the local search is used to find the likely object configuration and the complex observation model is summarized by local approximation around the likely configuration. The simplest case is to approximate the observation model by a single Gaussian function centered at the most likely object configuration. The tracking is then achieved analytically by the Kalman filter. In our implementation the Kalman filter requires just a few milliseconds per frame. For the "simple" tracking situations the results are comparable to the other tracking schemes. However, in the more challenging situations where the tracked object is occluded from time to time the performance of the Kalman filter degrades significantly. The degradation occurs for two reasons. The first reason is that only the previously estimated object state is used as the starting point for the local search. In case of occlusions and sudden movements it is difficult to find the most likely object configuration again and

the local search can end into some local maxima. To increase the chance to find the most likely object configuration we use a number of starting positions sampled from the prediction density. The second reason for the decreased performance is that in case occlusions and background clutter there might be a number of likely object configurations. The configurations may be associated with different hypothesis about the object trajectory and the correct trajectory can be disambiguated only after observing future data. Therefore we approximate each likely configuration locally by a Gaussian which leads to a Gaussian mixture approximation of the observation model. The approximate scheme known as Mixture Kalman filter can be used then for tracking. The computational costs for approximating the observation model increase proportional to the number of starting points for the local search. There are also some additional computation costs for performing the Mixture Kalman filter update steps. When using 5 starting points the Mixture Kalman filter in our implementation required 2-3 times more time per frame than the Kalman filter but the results greatly improve especially for the difficult tracking situations. Another common approach for approximate Bayesian filtering is to use a sampling scheme for example the bootstrap particle filter. Empirical results show that for small number of samples that are needed for real-time implementation the bootstrap filter suffers similar problems as the Kalman filter. The novel particle filter we propose is using the Gaussian mixture approximation of the observation model obtained using the local search as the proposal distribution. The results show that this particle filter is more robust then the bootstrap filter while the theoretical guarantees are retained [11]. We believe that the improvement in general depends on the efficiency of the local search and the goodness of the approximation. We expect the most benefit from an efficient search in case we have a highly dimensional state space, since it is well known that the performance of importance sampling quickly degrades in high dimensions unless an adaptive and accurate proposal distribution is used. The computation costs for the sample based approaches scale with the number of particles. The novel particle filter includes also the costs of approximating the observation model by the mixture. Empirically the new particle filter with proposal and 100 particles requires twice the time of the the mixture Kalman filter. The tracking results improve without compromising theoretical convergence properties that the mixture Kalman filter fail to have. On the other hand the mixture Kalman filter is a better choice if the high accuracy is less important than computation time.

References

[1] A.Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[2] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[3] B.Han, D.Comaniciu, Y.Zhu, and L.Davis. Incremental density approximation and kernel-based bayesian filtering for object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[4] B.Han, Y.Zhu, D.Comaniciu, and L.Davis. Kernel-based bayesian filtering for object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[5] P. Chang and J. Krumm. Object recognition with color cooccurrence histograms. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1999.

[6] R. Chen and J. Liu. Mixture kalman filters. *Journal of Royal Stat. Soc. B*, 62:493–508, 2000.

[7] R. Collins and Y. Liu. On-line selection of discriminative tracking features. *In Proc. International Conference of Computer Vision*, 2003.

[8] T.F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *Computer Vision Image Understanding*, 61(1):38–59, 1995.

[9] D.Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):564–575, 2003.

[10] D.Comaniciu, V.Ramesh, and P.Meer. Real-time tracking of non-rigid objects using mean shift. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:142–149, 2000.

[11] A. Doucet, N. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[12] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[13] F.Ennesser and G. Medioni. Finding waldo, or focus of attention using local color information. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(8):805–809, 1995.

[14] R. Fletcher. *Practical Methods of Optimization*. J. Wiley, 1987.

[15] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[16] J. Giebel, D. M. Gavrila, and C. Schnrr. A bayesian framework for multi-cue 3d object tracking. *In Proc. European Conference on Computer Vision*, 2004.

[17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proc. Part F, Radar and Signal Processing*, volume 140(2), pages 107–113, 1993.

[18] G.R.Bradski. Computer vision face tracking as a component of a perceptual user interface. *In Proc. IEEE Workshop on Applications of Computer vision*, pages 214–219, 1998.

[19] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 790–797, 2004.

[20] H.Zhang, W.Huang, Z.Huang, and L.Li. Affine object tracking with kernel-based spatial-color representation. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[21] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[22] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *In Proceedings of the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[23] K.Branson, V.Rabaud, and S.Belongie. Three brown mice: See how they run. *In Proc. IEEE International Workshop on VS-PETS*, 2003.

[24] K.Fukunaga and L.D.Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 2002.

[25] K-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[26] M.Fashing and C.Tomasi. Mean shift is a bound optimization. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(3):471 – 474, 2005.

[27] T. Minka. Expectation-maximization as lower bound maximization. *Tutorial note*, 1999.

[28] M.J.Black and A.D.Jepson. Eigentracking: Robust matching and tracking articulated objects using a view based representation. *Intrenational Journal Computer Vision*, 26(1):63–84, 1998.

[29] K. Nummiaro, E. Koller-Meier, and L.J. van Gool. An adaptive color-based particle filter. *Image Vision Computing*, 21(1):99–110, 2003.

[30] F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[31] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[32] R.Collins. Mean-shift blob tracking through scale space. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[33] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[34] S.T.Birchfield and S.Rangarajan. Spatiograms versus histograms for region-based tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[35] M. Swain and D. Ballard. Color indexing. *Intl. J. of Computer Vision*, 7(1):11–32, 1991.

[36] C. Yang, R. Duraiswami, and L. Davis. Efficient spatial-feature tracking via the mean-shift and a new similarity measure. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[37] A. Yilmaz and M. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(11):1531–1536, 2004.

[38] Z. Zivkovic and B. Krose. An EM-like algorithm for color-histogram-based object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

Data Fusion for Visual Tracking dedicated to Human-Robot Interaction

L. Brèthes†, F. Lerasle†‡, P. Danès†‡
{*lbrethes,lerasle,danes*}*@laas.fr*
†*LAAS - CNRS*
7 avenue du Colonel Roche, 31077 Toulouse, France
‡*Université Paul Sabatier*
118 route de Narbonne, 31062 Toulouse, France

***Abstract*— The interaction between men and machines has become an important topic for the robotics community as it can generalize the use of robots. In this context, advanced robots must integrate capabilities to interpret humans motion as well as persons gestures in order to perform tasks for the humans or in synergy with them. The purpose of this paper is to show a real-time system for face/hand tracking and hand gesture recognition in the particle filtering framework. We introduce mechanisms for visual data fusion within particle filtering to develop trackers combining in a novel way color and shape cues, skin blobs or frontal face detection. For the purpose of face tracking, the fusion of modalities based on color and shape allows to avoid noticeable drift, even possible subsequent loss in the worst case. For gestures interpretation, an extension is proposed to achieve in the tracking loop the recognition of the current hand posture and of its motion in the video stream. In both tracking scenarios, the combination or fusion of cues proves to be more robust in cluttered environments than any of the cues individually. The global performances of the proposed trackers and future works are also discussed.**

I. Introduction and framework

Man-machine interaction has become an important topic in the robotics community. In this context, advanced robots must integrate capabilities to detect humans presence in their vicinity and interpret their motion. Here, persons are just considered as some "passers by" and no direct interaction is intended. For an active interaction, the robot must also be able to interpret gestures performed by the tracked person. We focus here on communicative gestures to symbolize some referential actions for the robot.

The purpose of this paper is to show a real-time system for face/hand tracking and hand gesture recognition in the particle filtering framework. This formalism has been pioneered in the seminal paper [4] by Isard and Blake. A first reason for focusing on particle filtering as the tracking engine comes from its capability to work with the non-Gaussian noise models required to represent the cluttered environments. A second reason of such a framework is that it allows the information from different measurements sources to be fused in a principled manner. Although this fact has been acknowledged before, it has not been fully exploited within a visual tracking context. Data fusion with particle filters has been mostly confined to skin color and shape cues inside and around simple silhouette shapes [5], while a host of cues (such as motion, color, sound) are sometimes available to increase the reliability of the tracking [11]. In [2], we proposed a preliminary approach based on contours (describing the shape) and skin regions segmentation to track faces and recognize hand configurations in video streams. The integration of skin blobs segmentation on board of our Rackham robot dedicated to H/R interaction (figure 1) showed that its behavior is greatly influenced by the variability of the environment itself (*e.g.* heavy cluttered background) and by the viewing conditions changes in such a mobile robot context. Skin blobs segmentation must be used cautiously and only when necessary.

Fig. 1. Our robot Rackham

Moreover, this cue as well as frontal face detection introduced in [2] are said intermittent because they are unefficient when the person turns back to the camera. This intermittent nature makes them candidate for the design of detection modules, efficient proposal distribution and particle filter initialization as depicted hereafter.

Color distribution on image patches describing the target are proved to be remarkably persistent and robust to changes in pose and illumination [11]. However, this cue remains prone to ambiguity with regard to false alarms characterized by a color distribution similar to that of the region of interest (ROI). These ambiguities can be drastically reduced by introducing shape cues, as human limbs to be tracked are known *a priori* so that silhouette models can be learnt beforehand (figure 5).

The remainder of the paper is organized as follows. Section II briefly outlines the well-known particle filtering formalism and alternative schemes when information from multiple measurement sources are available. Section III presents four cues we aim to combine in trackers dedicated to H/R interaction: frontal face detection, skin blobs detection, contours (shape) and color. Applications of face tracking and gestures recognition (both static and dynamic) are presented in sections IV and V under a variety of conditions/scenarios. Finally, section VI summarises our contribution and opens the discussion for future works.

II. Condensation formalism and data fusion

A. The "Condensation" algorithm

The "Condensation" algorithm —for "Conditional Density Propagation"— is a particle technique for the estima-

tion of the state vector of a nonlinear Markovian system submitted to possibly nonGaussian random inputs [1], [7]. The aim is to recursively estimate the *a posteriori* probability density of the state vector x_k at time k conditioned on the knowledge of past measurements.

Let z_1^k term the available measurements from time 1 to k. At each time k, the probability density function (pdf) $p(x_k|z_1^k)$ is depicted by a set of particles $x_k^{(i)}$ —which are samples of the state vector— affected by weights $w_k^{(i)}$. The idea is to get

$$p(x_k|z_1^k) \approx \sum_i w_k^{(i)} \delta(x_k - x_k^{(i)}), \qquad (1)$$

i.e. to approximate random sampling from the pdf $p(x_k|z_1^k)$ by the selection of a particle with a probability equal to its associated weight. All the difficulty thus lies in the way the particles and their weights are defined all along the estimation process. Moments of the *a posteriori* distribution can then be approximated through the formula $E\big(g(x_k|z_1^k)\big) = \sum_{i=1}^{N_i} w_k^{(i)} g(x_k^{(i)})$.

The estimator initialization consists in the definition of a set of weighted particles which can describe the initial prior $p(x_0)$. Then, starting from a set of weighted particles $\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}$ associated with the filtering density $p(x_{k-1}|z_1^{k-1})$ at time $k-1$, the computation of the particles set associated to the filtering density at next time k proceeds in three steps.

The first step consists in the resampling, within the set $\{x_{k-1}^{(i)}\}$, of N_i new particles $x'^{(i)}_{k-1}$. This resampling is guided by the weight values, in that $P(x'^{(i)}_{k-1} = x_{k-1}^{(i)}) = w_{k-1}^{(i)}$. So, particles associated to high weights $w_{k-1}^{(i)}$ may be duplicated, while low-weighted particles collapse. The consequent uniformly weighted particle set $\{x'^{(i)}_{k-1}, N_i^{-1}\}$ still represents $p(x_{k-1}|z_1^{k-1})$.

Then, each particle $x'^{(i)}_{k-1}$ is propagated between times $k-1$ and k by generating its successor from the pdf $p(x_k|x_{k-1} = x'^{(i)}_{k-1})$ relative to the hidden state vector dynamics. It can be easily shown that $\{x_k^{(i)}, N_i^{-1}\}$ describes the prediction density $p(x_k|z_1^{k-1})$.

Finally, the set of weighted particles associated to the filtering density at time k is determined by taking into account the measurement z_k. The Bayes rule shows that $\{x_k^{(i)}, w_k^{(i)}\}$ describes $p(x_k|z_1^k)$ as soon as each weight $w_k^{(i)}$ relative to $x_k^{(i)}$ is affected the value $p(z_k|x_k = x_k^{(i)})$, prior to a normalization of the $w_k^{(i)}$'s so that $\sum_{i=1}^{N_i} w_k^{(i)} = 1$.

The Figure 2 shows an example. Therein, each ellipse is centered on a particle and has a size related to its weight.

B. Application to visual tracking

Visual tracking can be stated as a filtering problem [7]. The state vector gathers a minimal set of variables relative to the target to be tracked. Its *a priori* dynamics, characterized by $p(x_k|x_{k-1})$, must be consistent with the admissible motions. The target is henceforth parametrized by some position, orientation and size parameters in the current frame. The state vector x_k at time k is made of their values at times k and $k-1$, so as to deal with temporal evolutions of the second order.

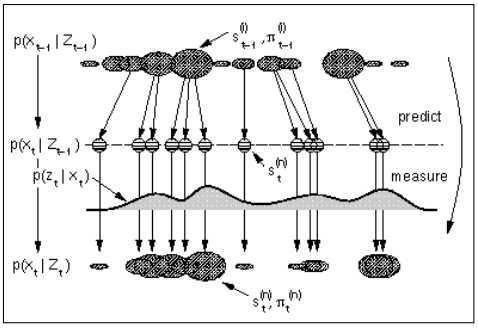


Fig. 2. Outline of "Condensation" algorithm. Blob centers represent particles and blob sizes depict the associated weights (from [7])

A target detection scheme —skin blobs, face detector,...— is applied to the initial frame in order to settle the initial state probability distribution of the tracker.

Several choices of the likelihood $p(z_k|x_k^{(i)})$ can be considered. It will be further assumed that when M measurement sources are used, they are independent conditioned on the knowledge of the state vector, so that $p(((z_1), \ldots, (z_M))_k|x_k^{(i)}) = \prod_{m=1}^{M} p((z_m)_k|x_k^{(i)})$.

C. Enhanced schemes

This paper also considers two enhancements of the original Condensation scheme: the ICondensation algorithm will be used for face tracking (section IV), while the Mixed-State Condensation will handle multiple gestures recognition (section V).

1) ICondensation: A deeper insight to the Condensation algorithm shows that the resampling step is necessary to avoid that after a few iterations, all but one particle have negligible weights. Indeed, such a degeneracy phenomenon cannot be avoided whatever the recursive particle filtering strategy. Yet, another fact can be used to limit this problem in addition to resampling, namely the choice of the importance —or proposal— density, *i.e.* of the way the particles are distributed in the state space [1].

Positioning the particles according to the stochastic state dynamics, as is the case in Condensation, isn't the optimal choice. Instead, the successor at time k of a particle $x_{k-1}^{(i)}$ should be drawn from an importance density $\pi(x_k|x_{k-1}^{(i)}, z_k)$ combining both the dynamics $p(x_k|x_{k-1}^{(i)})$ and the actual measurement z_k [3]. Notice that a systematic procedure is defined so as to update its weight accordingly.

The ICondensation algorithm [5] is a step towards this aim. In this approach, particles at time k can be drawn from a pdf of the form $\pi(x_k|z_k)$ according to the ROIs in the current frame. Practically, they can be selected in the vicinity of color blobs (section III-B).

However, if a particle drawn exclusively from the image data is inconsistent with its predecessor from the point of view of the state dynamics, the update formula leads to a small weight. In order to avoid this problem, the ICondensation implementation also draws some particles

following the original Condensation scheme and others using the prior distribution $p(x_0)$.

2) A Condensation algorithm for jump Markov systems: The Condensation algorithm can also be readily extended to tackle jump Markov systems, see the "mixed-state" version of [6].

Let l_k be a variable taking its values in a discrete set —typically a gesture index— and following a discrete-time Markov chain with known transitions probabilities T_{ij}. Assume that the state vector x_k and the measurement z_k obey a known jump Markov system such that

$$\begin{aligned} &p(x_k|x_{k-1}, l_k, l_{k-1}) = p(x_k|x_{k-1}, l_k) = p_{l_k}(x_k|x_{k-1}) \\ &\text{and } p(z_k|x_k, l_k, l_{k-1}) = p(z_k|x_k, l_k) = p_{l_k}(z_k|x_k). \end{aligned} \quad (2)$$

Stating $X_k = (l_k, x_k)$ enables to deal with such a system in the Condensation framework.

On the one hand, in the prediction step a particle $X_k^{(i)} = (l_k^{(i)}, x_k^{(i)})$ is sampled from the dynamics prior

$$\begin{aligned} p(X_k|X_{k-1}^{(i)}) &= p(l_k, x_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}) \\ &= p(x_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}, l_k)p(l_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}), \\ &= p_{l_k}(x_k|x_{k-1}^{(i)})T_{l_{k-1}^{(i)} l_k}. \end{aligned} \quad (3)$$

A straight way to perform this is to first sample the discrete index $l_k^{(i)}$ from the transition probabilities $T_{l_{k-1}^{(i)} l_k}$, and then draw $x_k^{(i)}$ from $p_{l_k^{(i)}}(x_k|x_{k-1}^{(i)})$.

On the other hand, the measurement update step involves the likelihood $p(z_k|X_k^{(i)})$.

A MAP estimate $\hat{l}_k$ at time k can be deduced, based on the sum of the weights of all the particles having the same discrete index at this time, and an estimate $\hat{x}_k$ follows:

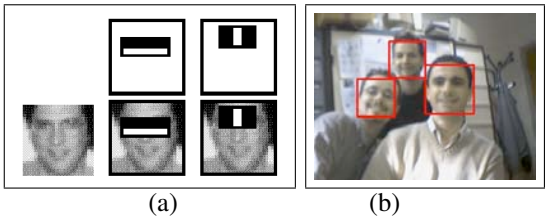
$$\begin{aligned} \hat{l}_k &= \arg\max_l \textstyle\sum_{i \in \Upsilon_l} w_k^{(i)}, \text{ with } \Upsilon_l = \{i : X_k^{(i)} = (l, x_k^{(i)})\} \\ \hat{x}_k &= \frac{\sum_{i \in \hat{\Upsilon}} w_k^{(i)} x_k^{(i)}}{\sum_{i \in \hat{\Upsilon}} w_k^{(i)}}, \text{ with } \hat{\Upsilon} = \{i : X_k^{(i)} = (\hat{l}_k, x_k^{(i)})\}. \end{aligned} \quad (4)$$

III. Measurement cues

We first focus on intermittent cues such as frontal face and skin regions detection. We then deal with persistent cues such as color and shape.

A. Frontal face detection

The method used for face detection was introduced by Viola *et al.* [12]. It is based on a boosted cascade of Haar-like features. These features are obtained by substracting the sum of the pixels lying inside the white rectangles from the sum of the pixels in the dark rectangles (Figure 3(a)). They enable the detection of relative darkness between



(a) (b)

Fig. 3. (a) Haar-like features overlaying on a training face, (b) example of face detection

eyes and nose/cheek or nose bridge. An over complete set of features is generated by scaling the Haar-like masks independently in vertical and horizontal directions. A cascade of classifiers is a degenerated decision tree where at each stage a classifier is trained to detect almost all frontal faces while rejecting a certain fraction of non-face patterns. This way, background regions are quickly discarded while focusing on promising frontal face-like regions (figure 3(b)).

B. Skin regions detection

Human skin colors have a specific distribution in color space. They can be clustered to form a feature space for segmentation. A color histogram model learnt offline is classically used to classify skin-like pixels. In our approach [2], a watershed-based segmentation is then applied on the labeled pixels to segment the skin regions.

Several color segmentation techniques have been used for skin blobs segmentation [8] using a skin pixel classification. However, in a mobile robot context, these are generally influenced by the variability of the environment clutters and the associated viewing conditions changes. Typically, overexposure (when the robot is close to a bay window) or underexposure (when the robot moves in a corridor) make more uncertain the separation of skin regions from background. Moreover, for cluttered environments, spurious close-to-skin colored regions can be sometimes segmented, for example wooden doors and desks (figure 4(b)). Yet, part of such false alarms can be eliminated regarding the aspect ratio of the region. Clearly, skin detection must be used cautiously and combined with other cues. Figure 4 shows two examples of correct and incorrect skin region segmentations.

(a) (b)

Fig. 4. Two examples: (a) correct segmentation, (b) incorrect segmentation due to clutter in background

C. Shape cue

The use of shape cue requires that the class of targets to be tracked is known *a priori* and that sufficiently precise silhouette models can be learned beforehand. Such conditions are met in human limbs tracking applications where coarse shape cues (of head or hand) can be used.

The aim here is to track faces and well-defined hand postures that represent a limited set of commands that the users can give to the robot. To use a simple view-based shape representation, face and hand are therefore represented by coarse 2D rigid models, *e.g.* their silhouette

contours, by means of splines [7]. These models, although simplistic, permit to reduce the complexity of the involved computations and remain discriminatory enough to track a set of known hand postures in complex scenes as will be shown later. Examples of these models are presented in figure 5.

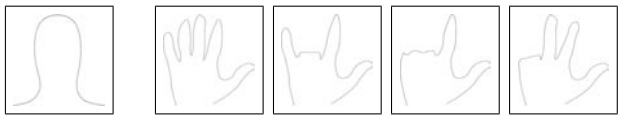


Fig. 5. Templates for face or hand configurations (depending on the number of open fingers)

In the particle filter measurement update step, each sample is classically given a likelihood that depends on the sum of the squared distances between model points and corresponding image points [7]. The model points are chosen to be uniformly distributed along the spline. The corresponding ones are found by searching in the image for color edge points which lie on the spline normals that pass through these points (figure 6).

Fig. 6. Measurement model for shape cue (from [7])

Color gradient can be estimated in various ways. It can be computed either as a combination of gradients issued from each channel separately, or as a vector to make full use of color information. We follow this last principled way (see [2] for more details).

Unfortunately, for cluttered background, using only shape cue for template fitting is not sufficient, as irrelevant contours may attach the tracker. Moreover, due to unfavourable illumination, the target contour may not be as prominent as expected, while color cues are known to be more robust to such lighting conditions.

D. Color distribution on image patches

1) Basics: For the color distribution modeling, we use independent normalized histograms computed in the RGB space. We denote the B-bin reference histogram model in channel $c \in \{R, G, B\}$ by $h^c_{ref} = (h^c_{1,ref}, \ldots, h^c_{B,ref})$. The color distribution $h^c_x = \left(h^c_{1,x}, \ldots, h^c_{B,x}\right)$ of a region B_x corresponding to any state x is computed by

$$h^c_{j,x} = c_H \sum_{u \in B_x} \delta_j(b^c_u), j = 1, \ldots, B, \qquad (5)$$

where $b^c_u \in \{1, \ldots, B\}$ denotes the histogram bin index associated with the intensity at pixel u in channel c of the color image z^C, δ_a terms the Kronecker delta function at a, and c_H is a normalization factor ensuring that $\sum_{j=1}^{B} h^c_{j,x} = 1$.

The color likelihood model must be defined so as to favor candidate color histograms close to the reference histogram. A popular measure between two distributions $h_1 = \{h_{j,1}\}_{j=1,\ldots,B}$ and $h_2 = \{h_{j,2}\}_{j=1,\ldots,B}$ is the Bhattacharyya coefficient [9]:

$$D(h_1, h_2) = (1 - \sum_{j=1}^{B} \sqrt{h_{j,1}.h_{j,2}})^{1/2}$$

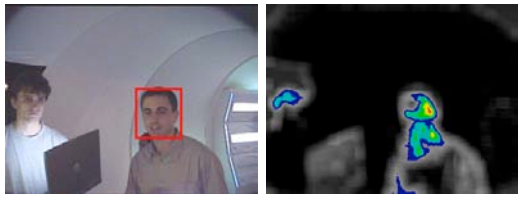
The smaller D is, the more similar the distributions are. Finally, the likelihood of a state x when faced to z^C is given by

$$p(z^C|x) \propto \exp(- \sum_{c \in \{R,G,B\}} D^2(h^c_x, h^c_{ref})/2\sigma^2_C).$$

If the tracked region contains different patches of distinct colors, e.g. the face and clothes of a person, the histogram-based modeling will capture them. It suffices to split the ROI into subregions, each with its own reference color model [10]. We consider the partition $B_x = \bigcup_{p=1}^{N_R} B_{p,x}$ associated with the set of reference histograms $\{h^c_{p,ref} : c \in \{R, G, B\}, p = 1, \ldots, N_R\}$. By assuming conditional independence of the color measurements within the different subregions defined by the state x, the multi-region color likelihood becomes:

$$p(z^C|x) \propto \exp(- \sum_{c \in \{R,G,B\}} \sum_{p=1}^{N_R} D^2(h^c_{p,x}, h^c_{p,ref})/2.\sigma^2_C)$$

where the histogram $h_{p,x}$ is collected in the region $B_{p,x}$. The histogram based definition of the color likelihood is illustrated in figure 7.



(a) (b)

Fig. 7. (a) ROI, (b) color likelihood of the location only (scale factor fixed)

2) Model update: Illumination conditions, out-of-plane rotated faces, visual angle as well as camera parameters are known to be difficult to handle as they may lead to an inaccurate tracking or a complete loss of lock. To overcome these appearance changes, we update the target model during slowly changing image observations. This update is made according to

$$h_{ref,k} = (1 - \alpha).h_{ref,k-1} + \alpha h_{E[x_k]}$$

where k terms the frame time and α weights the contribution of the mean state histogram $h_{E[x_k]}$ *w.r.t* the target model $h_{ref,k-1}$. The contribution of a specific frame decreases exponentially in time. The channel index c and bin index j have been omitted for compactness reasons.

IV. Application to face tracking

Color-based filtering schemes enable the robust tracking of targets undergoing complex changes in shape and appearance. Unfortunately, due to the model updating, noticeable drifts or even loss of target can be observed [11]. The robustness of the tracker to drifts and color clutters can however be increased by incorporating multi-patches color models and by fusing color and shape cues in the measurement model. This is illustrated in snapshots from a sequence of 300 frames (figure 8).

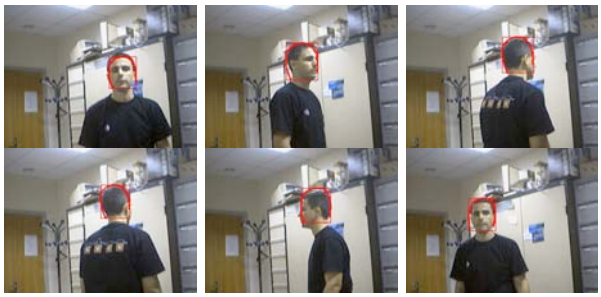


Fig. 8. Face tracker fusing color and shape cues (images $58, 129, 142, 159, 188, 234$): better postioning on the target, weak drift

Moreover, considering multi-patches (on face and especially clothes) of distinct color distribution makes the tracker keep focusing on the current target even if several persons enter in the view field of the camera (figure 9).

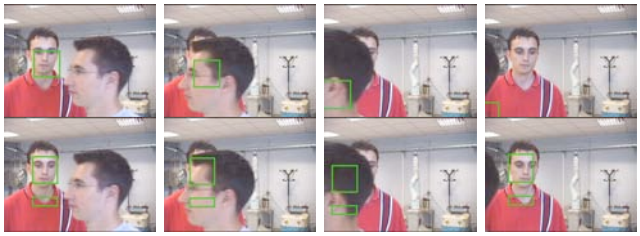


Fig. 9. Influence of the multi-part color model in the tracker images $246, 254, 261, 268$): (top) with a single part, the tracker locks to a wrong target in the foreground. (bottom) with multi-part model, the tracker keeps locked onto the ROI even after an occlusion

As for the computational cost, a non-optimized implementation tracks regions of about 20×20 pixels at a rate of $50\ fps$ with $M = 200$ particles on a $3GHz$ Pentium IV.

Last improvements concern the tracking of targets undergoing rapid motion, persistent occlusion or (re)-apparition in the scene. The aim is to make the tracker robust to such events, which generally would cause target loss.

In the Condensation scheme, the state evolution model is used to sweep the image ROIs in which the target is predicted to lie. This precludes any re-initialization of the tracker when the target can be anywhere in the image. The ICondensation depicted in section II fulfills such problems by using proposal distributions based on face detection or skin blobs detection (low-level) and by fusing color and shape cues in the likelihood model (high level).

Let B be the number of detected blobs. Following [5], the centroid of each blob is computed as a coordinate b'_i in the original image, and a 2D importance function π is defined by the Gaussian mixture

$$\pi(x_k|z_k) = \sum_{i=1}^{B} \delta_i . \mathcal{N}(b_i, \Sigma_B)$$

where $b_i = b'_i + \bar{x_B}$, and $\bar{x_B}$ and Σ_B are the mean and covariance respectively of the offset from the blob (or face) position to the centroid of the contour describing the face. These parameters are learned offline by using a contour tracker and comparing the output of skin blobs (or face) detection with the centroid of the tracked contour.

V. Application to gestures recognition

In [2], we proposed a preliminary approach to the recognition of the current hand posture (figure 5) and the automatic switching between multiple templates in the tracking loop. For a richer interaction, an extension of this tracker is proposed so as to handle multiple canonical motion models as classifiers for gesture recognition.

The bayesian mixed-state framework depicted in section II is well-suited to manage hand motion and configuration models in video streams. Indeed, it suffices to augment the state vector by two discrete variables respectively indexing the configuration and the motion type. So far, these indexes have been assumed mutually independent, and evolve over time according to distinct transition probabilities matrices. A further step will consist in defining these switching probabilities from the predefined interaction language.

As aforementioned, color and shape modalities are mixed in the measurement model while the extension to multi-part color modelling is efficient to discriminate between configurations. This last issue is achieved within our color model by splitting the tracked region into subregions corresponding to the palm and fingers, and by considering a single reference color model which is related to the palm in the previous frame. Local Bhattacharyya distances on these ROIs can exhibit the presence or absence of open fingers, thus improving the discriminative power between templates associated to configurations. Pratically, the smaller the color discrepancy between a given ROI and the reference model, the higher is the probability that an open finger is located inside this ROI.

Regarding the experimentations, we consider two main scenarios. In the first one, the behavior of the tracker is illustrated when using only color cue. In the second one, we fused color and shape cues to recognize both hand postures and motion models.

A. Considering color cue only

With no assumption regarding hand silhouette templates, that is, considering only color cue, a moving hand can be tracked with a reasonable accuracy as shown in the sequence of figure 10. In this sequence, the contour is drawn in green (resp. red) during roughly horizontal (resp.

vertical) motions and in blue if the hand remains stationary. The classification of motion by model-switching is accurate in most cases and the tracker runs at about $60\ Hz$.

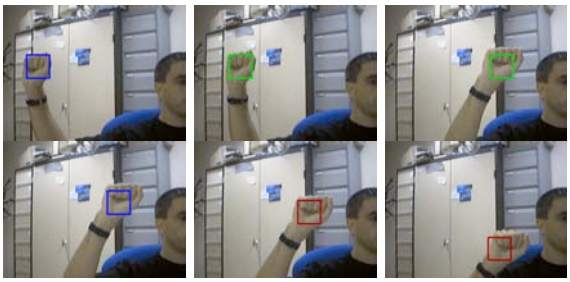


Fig. 10. Hand tracker based on only color distribution: images $29, 32, 46, 47, 56, 64$

B. Fusing color and shape cues

Figure 11 shows a few snapshots from a sequence of 200 frames, where the hand moves in front of a cluttered background while its posture and global motion changes. In this sequence, the contour is drawn in pink (resp. green) during roughly horizontal (resp. vertical) motions and in blue if the hand remains stationary.

Fig. 11. Fusing color distribution and edges in the hand tracker: images $26, 32, 44, 45, 58, 71, 93, 104, 152$

Recognition results are compared with a ground truth for both hand postures and motion models. While the hand posture is correctly determined in most frames (close to 99%), the motion model is more often misclassified: about 75% for vertical or horizontal models, only 65% for the stationarity (due to hand shivering).

VI. Conclusion and future works

In this paper we introduced mechanisms for data fusion within particle filtering to develop trackers combining color, edges based cues, eventually skin blobs or frontal face detection in a novel way. Being the most persistent, the two first cues were used as the mains cues for tracking. The two last ones, logically intermittent, act in detection and initialization modules for the particle filter.

For face tracking purpose, the fusion of color distribution and edges-based modalities allows to avoid noticeable drift and possible subsequent loss, experienced sometimes by considering these cues individually. Considering multiple subjects in the view field, multi patches of distinct color distribution (such as the face and clothes) allows the tracker to keep focusing on the current target. For gestures tracking/recognition purpose, our tracker was adapted to track multiple templates (representing hand postures) and associated motion models. In both tracking scenarios, the combinaison or fusion of cues proved to be more robust than any of the cues individually. Videos of the diffrent trackers are available at www.laas.fr/∼lbrethes/icra05/

Furthermore, we want to involve the fusion of other information such as sound or motion intermittent cues (which are less prone to clutter) and adapt our tracker to be able to track multiple targets simultaneously. The multiple target tracking could also be applied to the two-handed gestures which is of great interest.

VII. Acknowledgements

The work described in this paper was partially conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactons on Signal Processing*, 50(2):174–188, February 2002.

[2] L. Brèthes, P. Menezes, F.Lerasle, and M. Briot. Face Tracking and Hand Gesture Recognition for Human-Robot Interaction. In *Int. Conf. on Robotics and Automation (ICRA'04)*, pages 1901–1906, 2004.

[3] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Cambridge University Department of Engineering, 1998.

[4] M.A. Isard and A. Blake. CONDENSATION-Conditional Density Propagation for Visual Tracking. *Int. Journal of Compution Vision (IJCV'98)*, 29(1):5–28, 1998.

[5] M.A. Isard and A. Blake. Icondensation: Unifying low-level and high-level Tracking in a Stochastic Framework. In *European Conf. on Computer Vision (ECCV'98)*, pages 893–908, 1998.

[6] M.A. Isard and A. Blake. A Mixed-state Condensation Tracker with Automatic Model-switching. In *Int. Conf. on Computer Vision (ICCV'98)*, pages 107–112, Bombay, 1998.

[7] M.A. Isard and A. Blake. Visual Tracking by Stochastic Propagation of Conditional Density. In *European Conf. on Computer Vision (ECCV'96)*, pages 343–356, Cambridge, April 1996.

[8] M.J. Jones and J.M. Rehg. Statistical Color Models with Application to Skin Detection. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, pages 274–280, 1999.

[9] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptative color-based particle filter. *Journal of Image and Vision Computing*, 21:90–110, 2003.

[10] P. Pérez, J. Vermaak C. Hue, and M. Gangnet. Color-based probabilistic tracking. In *European Conf. on Computer Vision (ECCV'98)*, pages 661–675, 2002.

[11] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *IEEE*, 92(3):495–513, 2004.

[12] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, 2001.

Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP*

Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann
IAIM, Institute of Computer Science and Engineering (CSE)
University of Karlsruhe(TH)
Karlsruhe, Germany
{*knoop,vacek,dillmann*}*@ira.uka.de*

***Abstract*— This paper describes a new approach for modeling joints in an articulated 3d body model for tracking of the configuration of a human body. The used model consists of a set of rigid generalized cylinders. The joints between the cylinders are modeled as artificial point correspondences within the ICP tracking algorithm, which results in a set of forces and torques maintaining the model constraints. It is shown that different joint types with different degrees of freedom can be modeled with this approach.**

Experiments show the functionality and robustness of the presented model.

***Index Terms*— Tracking, ICP, 3D body model, Joint Constraints**

I. Introduction

Robots that are meant to cooperate closely with humans, and especially with untrained persons who are not familiar with the domain of robotics, need a deep understanding of the intentions, activities, actions and movements of their human interaction partner. This claim evolves from several facts:

On the one hand, the robot needs to be able to predict the global plan and intention of the human in order to plan its own actions within a cooperation context. Often, parts of this knowledge can be communicated explicitly by speech, gestures, or similar ways. But even if the global goal can be easily shared between human and robot, there might be multiple ways of performing the task in cooperation. Thus, an activity recognition and, based on this, an intention prediction is indispensable. Activity recognition must at least partly base on knowledge about detailed movements of the human body, which can be derived from a tracking system with appropriate models of the human.

On the other hand, the robot needs to plan its movements in a workspace that is shared between human and robot. This puts up high safety demands, including not only collision avoidance between human and robot, but also safe haptic interaction e.g. for handing over objects, or shared objects and tool manipulation. Therefore, an observation and prediction of human movements is badly needed on a humanoid robot which is designed to work together with humans. This also implies the need for a system that is able to track human motion.

To yield reasonable results, a tracking system for human motion has to exhibit an expedient model of the human body. Several models have been proposed in literature, some of which are discussed in section II. This body model has to fulfill several requirements: The description should be adaptable to different persons, it should be possible to incorporate known constraints to reduce search space, and the granularity has to be on the one hand high enough to deliver sufficient results, but on the other hand coarse enough to allow for online tracking.

This paper presents an approach for an articulated 3d human body model, and especially proposes a new joint model. This joint model provides the possibility to model "soft" constraints by applying an elastic band approach. It can be easily integrated in tracking approaches, as it will be shown with the *Iterative Closest Point* (ICP) algorithm. First experiments and results will be presented.

Section II gives a short overview on other works concerning human models for tracking and activity recognition, and section III presents the framework used for the described work. The proposed human and joint model are depicted in section IV and section V. Section VI presents experiments and results.

II. State of the Art

For observation and tracking of human movements and prediction of intentions, many different sensors and models have been used, including invasive sensors like magnetic field trackers (see e.g. [1], [2]) that are fixed to a human body. Within the context of human-robot-interaction in everyday life, this approach is not feasible; non-invasive tracking approaches must be applied. Most of these are based on vision systems, or on multi-sensor fusion (see [3]).

Tracking of humans and human body parts using vision is investigated by a lot of research groups and several surveys exist (see [4], [5], [6], [7]. Hence, there is a big variety of methods ranging from simple 2D approaches like skin color segmentation (e.g. [8]) or background subtraction techniques (e.g. [9]) up to complex reconstructions of the human body

*The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

pose. [10] shows how to learn the appearance of a human using texture and color.

Estimating the 3D pose of a human has become popular in recent years due to improved and new sensors and also due to available computing power making complex calculations feasible. Sidenbladh [11] used a particle filter to estimate the 3D pose in monocular images. Each particle represents a specific configuration of the pose which is projected into the image and compared with the extracted features. In [12] a *shape-from-silhouette* approach is used to estimate the human's pose.

An ICP-based approach for pose estimation is shown in [13]. The authors use cylinders to model each body part. In [14] the same authors show how they model joint constraints for their tracking process. However, it seems that the effect of the ICP is partially removed if the constraints are enforced. Nevertheless, parts of the work described in this paper are based on the work of Demirdlian et al. (see [14], [13]).

III. Used Framework

In the following, a brief overview is given on the sensors (see section III-A) and the Iterative Closest Point algorithm (see section III-B).

A. Sensor Data

The algorithm is based on a 3D point-to-model matching performed by the ICP. To be able to match the geometric model to a point cloud, it is necessary to obtain 3D point measurements.

Two different sensors are used for this purpose: A Time-of-Flight (ToF) camera and a standard stereo camera head with depth data reconstruction.

The *Swissranger* ToF camera uses a resolution of 160×124 pixels. Its output consists of a dense depth image and an intensity image. The depth range is configured to $0.5m \leq range \leq 7.5m$, the accuracy lies within a few centimeters. Intensity data is not used within the current context, as the intensity image has very low resolution and high noise due to the sensor concept.

The stereo camera (*mega-d* from *Videre Design*) has a maximum resolution of 1024×960, but for the given tracking purpose, only 320×240 is used. The disparity image is computed based on a calibration obtained offline.

An example scene, the corresponding disparity image from the stereo vision system and the depth image from the ToF-sensor can be seen in fig. 1.

B. Iterative Closest Point Algorithm

In the following section, a short introduction into the principles of the ICP (*Iterative Closest Point*) algorithm is given. The idea is that one has two indexed sets of the same points in two different coordinate systems and wants to calculate the translation $\vec{t}$ and rotation $\mathbf{R}$ which transform the first set into the second. For person tracking, the first set corresponds to the data points of the sensor and the second set corresponds to points on the surface of a rigid body. Following [15] the first set is denoted $P = \{\vec{p_i}\}$, the second one $X = \{\vec{x_i}\}$. Both sets have the same size with $N_x = N_p = N$ and each point $\vec{p_i}$ corresponds to point $\vec{x_i}$.

Having six degrees of freedom, at least three points are needed to calculate the rotation and the translation. Because the sensor data is always corrupted with noise, no exact solution exists. Instead, the problem is transformed into the minimization of a sum of squared distances:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}) + \vec{t} - \vec{p_i}||^2 \tag{1}$$

Being $\vec{\mu_p}$ and $\vec{\mu_x}$ the mean value of P and N respectively and setting $\vec{p_i}' = \vec{p_i} - \vec{\mu_p}, \vec{x_i}' = \vec{x_i} - \vec{\mu_x}$ and $\vec{t}' = \vec{t} + \mathbf{R}(\vec{\mu_x}) - \vec{\mu_p}$ equation 1 becomes:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}') - \vec{p_i}' + \vec{t}'||^2 \tag{2}$$

Evaluating yields:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \Big(\sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}') - \vec{p_i}'||^2 - 2\vec{t}' \cdot \sum_{i=1}^{N} (\mathbf{R}(\vec{x_i}') - \vec{p_i}') + N||\vec{t}'||^2 \Big) \tag{3}$$

The first part is independent of $\vec{t}'$, the second part reveals to zero, therefore the function becomes minimal if $\vec{t}' = 0$. Transforming back gives:

$$\vec{t} = \vec{\mu_p} - \mathbf{R}(\vec{\mu_x}). \tag{4}$$

Having the optimal translation (and thus $\vec{t}' = \vec{\mathbf{0}}$), equation 2 becomes:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}') - \vec{p_i}'||^2 \tag{5}$$

Considering $||\mathbf{R}(\vec{x_i}')|| = ||\vec{x_i}'||$, the equation can be written as:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \Big(\sum_{i=1}^{N} ||\vec{x_i}'||^2 - 2 \cdot \sum_{i=1}^{N} \mathbf{R}(\vec{x_i}') \cdot \vec{p_i}' + \sum_{i=1}^{N} ||\vec{p_i}'||^2 \Big) \tag{6}$$

Maximizing

$$\sum_{i=1}^{N} \mathbf{R}(\vec{x_i}') \cdot \vec{p_i}' \tag{7}$$

gives the optimal rotation. See [16] for details.

In the beginning of this section it has been assumed that two indexed sets P and N exist with $\vec{p_i}$ corresponding to $\vec{x_i}$. In reality only an unordered list of data points from the sensor(s) and a geometrical description of the body (the model points) can be accessed. Therefore in the first step the indexed lists have to be constructed. This is done by calculating for each

Fig. 1. Input data. 2d image (left), disparity image (middle), 3d image (right)

data point $\vec{p_i}$ the *closest point* on the model giving $\vec{x_i}$. In the second step the optimal translation and rotation can be estimated and applied to the model. Afterwards the closest points have to be calculated again with the new position of the model in order to get the sum of squared distances between the data points and the model points.

The *Iterative Closest Point* works as follows:

1) For a given model and its data points calculate the closest points giving CP_0
2) Calculate the sum of squared distances between data points and model points giving $d_0(M, CP_0)$
3) Estimate rotation and translation and apply to the model
4) Calculate new set of closest point with new position of the model giving CP_i
5) Calculate the sum of squared distances between data points and model points giving $d_i(M, CP_i)$
6) If $d_{i-1}(M, CP_{i-1}) - d_i(M, CP_i) < \epsilon$ stop the iteration, otherwise go to step 3.

Note that computation of *closest point relations* is by far the most time consuming step in the ICP process, since it includes a set of geometric calculations for each data point in the point cloud.

IV. 3D Human Body Model

A 3D body model is used for the tracking system. Each body part is represented through a *degenerated cylinder* (see fig. 2 left). Top and bottom of each cylinder is described by an ellipse. The ellipses are not rotated to each other and the plains are parallel. In total, such a body is described by five parameters (major and minor axis of the bottom ellipse, major and minor axis of the top ellipse and the length of the cylinder). The coordinate system is placed in the center of the bottom ellipse with the x-axis along the major axis, the y-axis along the minor axis and z along the height of the cylinder.

The overall body model is built in a tree-like hierarchy starting with the torso as its root body part. Each child is described with a degenerated cylinder and the corresponding homogenous transformation matrix. Up to now the body model consists of ten body parts (torso, head, two for each arm and two for each leg) which is depicted on the right of fig. 2. It should be mentioned that this body model is not necessarily restricted to humans, also other bodies can be modeled like a snake or a rabbit.

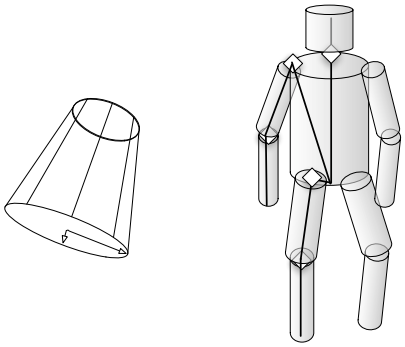


Fig. 2. Left: Degenerated cylinder, right: The overall human body model

A. 3D Human Model within ICP

For applying the ICP-algorithm to a 3D body model, two things have to be considered: First, the ICP algorithm has to be extended to an articulated model. Following the definitions in section III-B, it can only cope with two point clouds, which corresponds to one rigid body. This extension is done by processing each model part separately and adding the joints as constraints in each step.

Second, an appropriate closest point function has to be defined.

Fig. 3 shows the principle of estimating the position of the closest points corresponding to the data points. X_a shows a regular pair of points whereas X_b is a special case which needs to be handled separately.

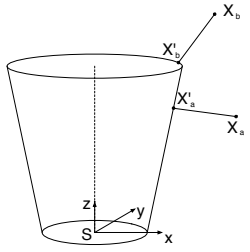


Fig. 3. Calculating the closest point on the degenerated cylinder

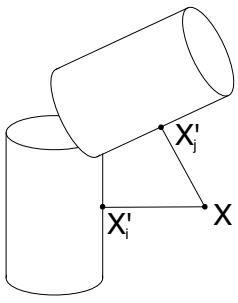


Fig. 4. One data point has to be assigned to two body parts

The 3D body model consists of several rigid bodies. Using the ICP requires partitioning of the input data points where

each data point is assigned to a special body part. Afterwards, the ICP can be applied on each body part. During the development, first results showed that it is not sufficient to create disjoint sets of points for each body part. Some points *share* two or more body parts instead, as can be seen in fig. 4.

Applying the ICP to each body part independently has one drawback which needs to be handled. Without additional constraints, it could happen that the single parts drift away and are no longer connected forming a body model. Therefore, the joints between the body parts need to be taken into account.

V. Joint Constraint Model

The joint model we propose is based on the concept of introducing elastic bands into the body model. These elastic bands represent joint constraints. For the ICP algorithm, these elastic bands can be modeled easily as artificial correspondences and will thus be considered automatically in each computation step.

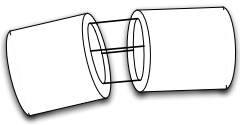


Fig. 5. Elastic bands as joint model

For each junction of model parts, a set of elastic bands is defined (see fig. 5). These relations set up corresponding points on both model parts. The corresponding points can then be used within the model fitting process to adjust the model configuration according to any sensor data input and to the defined constraints.

A. Joint Constraint Types

With this approach, different types of joints can be modeled. Looking at a model for the human body, different kinds of joints with varying degrees of freedom are required:

- **Universal Joints** have 3 full degrees of freedom. This joint type can be found e.g. in the shoulder. The upper arm can rotate up/down, forward/backward and around its main axis.
  Universal joints are modeled by a point-to-point correspondence (one elastic band) between both body parts with one point on each, see fig.6 a).
- **Hinge Joints** have one real degree of freedom, the others being almost fixed. This can be found e.g. in the human knee or elbow (only 1 DoF), or in the hip (1 real DoF, the other two existing, but highly restricted in motion).
  Hinge joints are modeled by a set of correspondences which are distributed along a straight line on both body parts. The same restriction can be achieved with correspondences only at each end of the line (two elastic bands), see fig.6 b).
- **Elliptic Joints** have all degrees of freedom highly restricted. An example of the human body is the neck (or the wrist): Motion is possible in all 3 degrees of freedom: Left/right, forward/backward, and turning. Each direction is very limited in range.
  Elliptic joints are modeled by a set of correspondences distributed along an ellipse on both body parts. This restriction can be achieved with correspondences on each end of the main axes of the ellipse (four elastic bands), see fig.6 c).

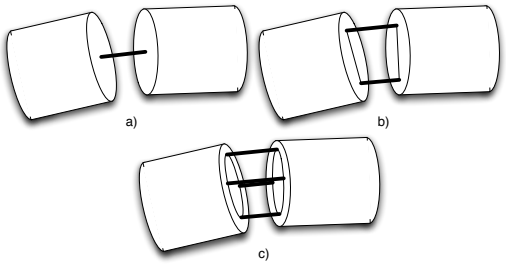


Fig. 6. Different joint type models. Universal Joint 3 DoF a), Hinge Joint 1 DoF + 2 restricted DoF b), Elliptic Joint 3 restricted DoF

Universal and hinge joints are special cases of the elliptic joint. For the hinge joint, one major axis of the ellipse is set to zero, resulting in a straight line. Setting both axes to zero produces a universal joint, because all correspondences are reduced to one point-to-point relation.

Following these definitions, each joint is modeled with a set of parameters describing the type of joint and its behavior. This parameter set consists of the major axes of the ellipse (a and b), its position and orientation on both body parts ($\vec{V}_i$ and $\vec{V}_{i+1}$), and the weight of the given correspondence (W). These parameters are now described in detail.

1) Major axes: The model type (universal, hinge, elliptic) and the valid range of each degree of freedom control the choice for the major axes sizes and ratio. Universal joints are modeled with both ellipse axes set to zero. For hinge and elliptic joints, the axis direction defines the rotation axis, and the axis length defines the stiffness of the other two rotational degrees of freedom. In fig. 6 b), rotational flexibility around the z-axis (perpendicular to the image plane) and around the symmetric axis of the cylinders is very limited due to the modeled joint.

2) Position and orientation: Position and orientation of the point-to-point, hinge or elliptic joint model with respect to both body parts define the connection between both parts.

3) Weight: If the joint model is used within a bigger tracking framework (see section V-B), the elastic bands can be used as correspondences which are included as tracking constraints. The use of measured correspondences together with artificial ones puts up the need for correct weighting strategies between input and model constraints.

To incorporate this, each joint model can be weighted with respect to measured input. This parameter is then used within the model fitting algorithm to balance between measured input and joint constraints. The weight parameter is defined in relation to the number of 'natural' correspondences to keep

the ratio between measurements and constraints.

Increasing the weight of a joint model tightens the coupling between both model parts, by decreasing the coupling becomes looser. For hinge and elliptic joints, higher weight also increases stiffness of a kinematic chain. This makes sense especially for joints like a human neck or wrist which shows a very tight coupling and very limited angular range.

The proposed joint model provides a "soft" way to restrict the degrees of freedom for the model parts. It additionally provides means to control the "degree of restrictiveness" for each DoF in a joint. Applying elastic strips is tantamount to introducing a set of forces which hold the model parts together. The connected points and magnitude define the joint behavior.

The soft joint model can e.g. be applied to the joint between human pelvis and thigh: While in forward/backward direction the movement is almost unrestricted, there is a high restriction for the left/right movement. But still a small movement is possible, and a 1 DoF model would not be sufficient.

Nevertheless, it is possible to model plain bending joints with a hinge joint model with one large axis.

B. Joint Model within the ICP

One of the main advantages of this joint model is that it can be very easily integrated in tracking algorithms. The joint model is added as a second data source, which adds correspondences between model and real world.

Introducing the joint model correspondences in the ICP framework (see section III-B) is done by transforming the elastic band constraints into artificial input points according to the following rules:

- For each correspondence with the weight W, W artificial point pairs are generated.
- The artificial point pairs have to be added to the correspondence list after computation of the *Closest Point Relations* (see section III-B, step 1).
- Because each body part is processed separately, each joint model has to be added twice, once to each associated body part.
- Each of the generated point pairs represents one point on the model and the associated artificial data point. So each pair has to be added to one body part as *Model - Data* and to the other *Data - Model* relation to retrieve the desired forces (from the elastic bands) on both model parts. These forces then try to establish the modeled connections.
- The artificial correspondences are recalculated in each ICP step.
- The chosen weight of each joint depends on the desired stiffness of the model. To always achieve the same stiffness during tracking, the ratio between measured and artificial point relations has to be constant. This means that the number of generated artificial points for one body part in each step depends on the quantity of measurements for this part of the model. The generated relations are linearly scaled with the number of measurements.
- From our experience, the ratio r between measured and artificial points should be chosen as approximately $0.4 \leq r \leq 0.7$. This gives enough cohesion within the model without implying to hard and static relationships.

It is important to note that the introduction of multiple identical correspondences within the ICP does not increase computation time with the order of point counts (like a set of different measured points would). The only additional effort consists of one multiplication of the resulting point matrix (4×4) with the scalar weight W.

VI. Experiments and Results

The framework described in this paper has been implemented and tested successfully on real data from the two sensors described in section III-A. Three example sequences are shown in fig. 7.

The first row in fig. 7 shows some frames (left data + model, right model only) of a sequence with a person performing rolling movement with the forearms (like a football player signaling substitution), with the arms periodically occluding each other. Two conclusions can be drawn from this sequence: First, the tracking with 3D data works not only on fully visible body parts, but also on partially or completely occluded parts. The forearms are partially occluded during the movement, but are still satisfactorily tracked. Second, the joint model also helps in tracking of occluded parts like the upper arm. Information on the torso and the forearm, both visible, determines the position and orientation of the upper arm. Another advantage of the given joint model is not visible at first sight: The orientation of the cylinders in symmetry direction (height axis) can not be extracted from measurements. Still, the joint model determines the correct pose of the cylinder (except for singularity poses) from the body pose. This can be helpful for other algorithms like texture mapping or matching etc.

The second and third row in fig. 7 show a comparison of the ICP on a body model with (2^{nd} row) and without (3^{rd} row) the hinge and elliptic joint. On the left, the ToF camera is used, on the right, the disparity data is used as input. It can be seen easily that the additional model knowledge improves tracking performance significantly. In the left two images, one arm is lost during tracking without joint restriction, while it tracked well with included joint models. The disparity based sequence shows the performance of the joint model: The joint model reveals its strength especially when the input data is noisy and defective.

The whole tracking framework does not run in realtime so far; a frame-rate of 3 to 5 fps can be processed on a standard PC.

VII. Conclusions and Future Work

This paper has presented a new approach for modeling joint constraints in an articulated 3D body model. This approach uses elastic bands and provides three basic joint types: Universal, hinge and elliptic joints. The model can be easily integrated into a tracking algorithm and has been successfully tested on real data within an ICP tracking framework. The model

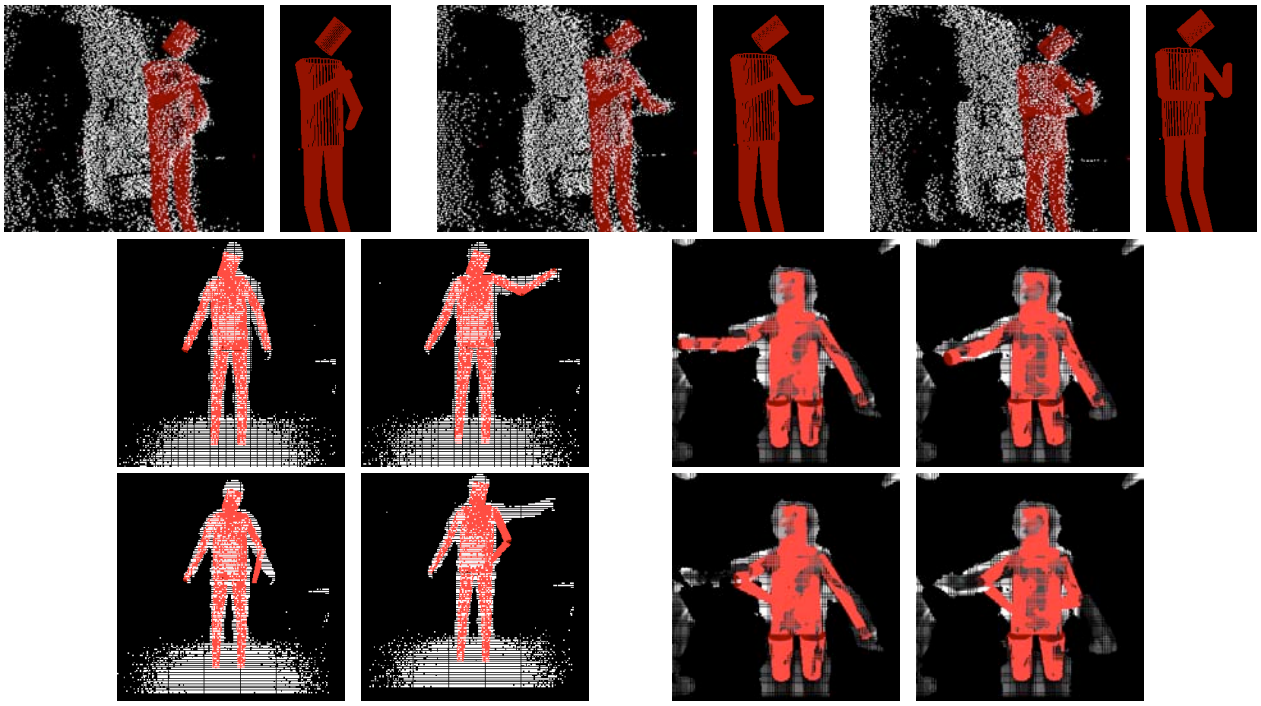


Fig. 7. Tracking results. First row: Tracking of rolling arms (data + model). Second and third row: Comparison of hinge/elliptic joint model (2^{nd} row) vs. simple universal joint model (3^{rd} row) with ToF data (left) and disparity image data (right)

constraint knowledge has been integrated similar to any other sensor data which gives information on the human pose.

Further research on tracking of human poses will mainly aim in three directions: First, the whole framework must be able to run in realtime to obtain robust online tracking. Second goal is to also explicitly include the valid angular range into the joint constraints by modifying the input data for the tracking algorithm, and third, we aim at integrating other sensory data into the tracking algorithm by generating artificial correspondences for the ICP in the same way the model constraints have been added. This leads to a generic framework for fusion of tracking algorithms and model knowledge.

REFERENCES

[1] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann, "Observation in programming by demonstration: Training and execution environment," in *Proceedings of Third IEEE International Conference on Humanoid Robots, October 2003, Karlsruhe*, Karlsruhe and Munich, Germany, 2003. [Online]. Available: http://wwwiaim.ira.uka.de/index.php/site/files/liste_veroeffentlichunge%n/1095668130/humanoids2003llncsFinal.pdf

[2] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm," in *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

[3] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer, "Multi-modal anchoring for human-robot-interaction," *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, vol. 43, no. 2–3, pp. 133–147, 2003.

[4] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, pp. 428–440, 1999. [Online]. Available: citeseer.ist.psu.edu/aggarwal99human.html

[5] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[6] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231–268, 2001.

[7] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003. [Online]. Available: http://nlpr-web.ia.ac.cn/english/irds/papers/wangliang/PR%20.pdf

[8] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, and G. Sagerer, "Improving adaptive skin color segmentation by incorporating results from face detection," in *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN)*. Berlin, Germany: IEEE, September 2002, pp. 337–343.

[9] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 257–267, 2001. [Online]. Available: citeseer.ist.psu.edu/bobick01recognition.html

[10] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *Computer Vision and Pattern Recognition*, vol. 2, 18-20 June, 2003, pp. II–467–II–474.

[11] H. Sidenbladh, "Probabilistic tracking and reconstruction of 3d human motion in monocular video sequences," Ph.D. dissertation, KTH, Stockholm, Sweden, 2001.

[12] G. K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *Computer Vision and Pattern Recognition*, 2003.

[13] D. Demirdjian and T. Darrell, "3-d articulated pose tracking to untethered diectic references," in *Multimodel Interfaces*, 2002, pp. 267–272.

[14] D. Demirdjian, "Enforcing constraints for human body tracking," in *2003 Conference on Computer Vision and Pattern Recognition Workshop Vol. 9*, Madison, Wisconsin, USA, 2003, pp. 102–109.

[15] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.

[16] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Optical Society of America Journal A*, vol. 4, pp. 629–642, Apr. 1987.

Excerpt from http://www.laas.fr/~lbrethes/KE2_2k5/

Particle filtering strategies evaluations

Several particle filtering strategies were evaluated in order to check which ones best fulfill the requirements of the considered H/R interaction modalities. The evaluated strategies are CONDENSATION, ICONDENSATION, the Auxiliary Particle Filter, the Rao-Blackwellized Subspace History-Sampling Sampling Importance Resampling (RBSSHSSIR) algorithm and the hierarchical particle filter (HIERARC).For the sake of comparisons, importance functions rely on dynamics or measurements alone (and are respectively noted DIF for Dynamics-based Importance Function and MIF for Measurement-based Importance Function), or combine both and are termed DMIF for Dynamics and Measurement-based Importance Function. Further, each modality has been evaluated on a database of sequences acquired from the robot in a wide range of typical conditions: cluttered environments, appearance changes or sporadic disappearance of the targeted person, jumps in her dynamics... For each sequence, the mean estimation error with respect to ``ground truth'', together with the mean failure ratio (% of target loss), were computed from several filter runs. Some associated figure plots, as well as some tracking scenarii, can be found here.

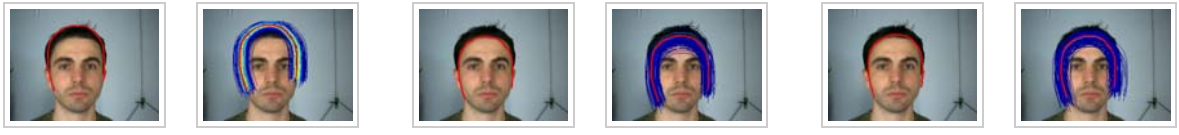
1. Short-range tracking

This modality combines motion and shape cues. An evaluation of various particle filtering strategies for this modality is presented below.

Nominal conditions

TOP

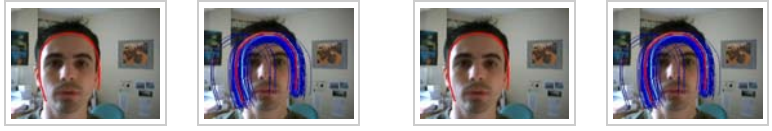
Video 1 : DIF strategy **Video 2 : MIF strategy** **Video 3 : DMIF strategy**



Cluttered background

TOP

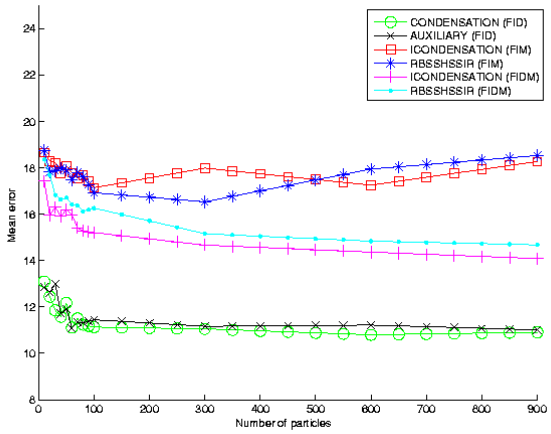
Video 4 : DIF strategy **Video 5 : MIF strategy**



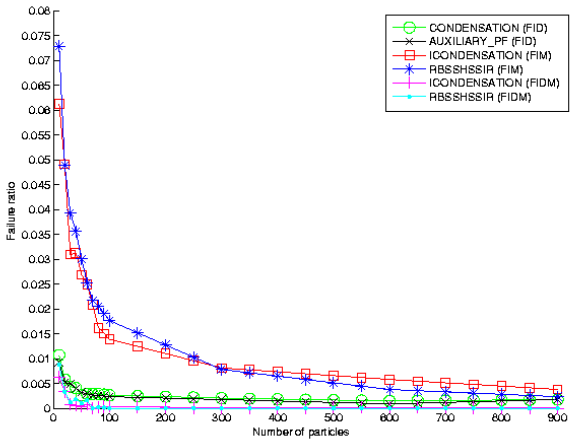
TOP

Mean error and failure ratio in cluttered background:

Mean error



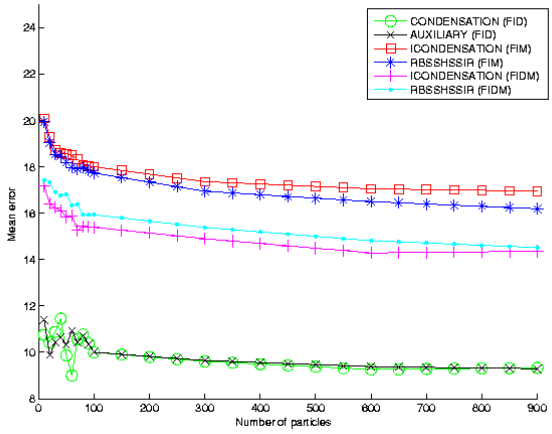
Failure ratio



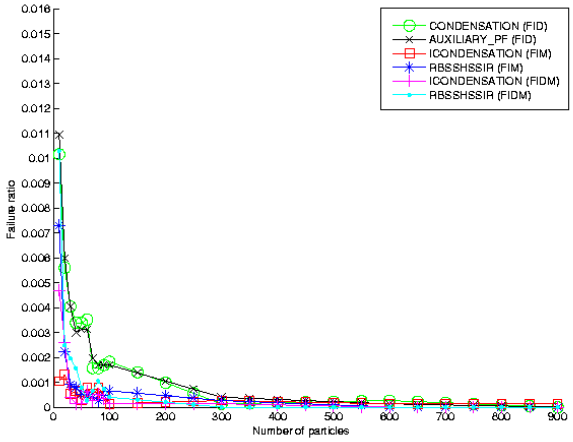
TOP

Mean error and failure ratio for in illumination:

Mean error



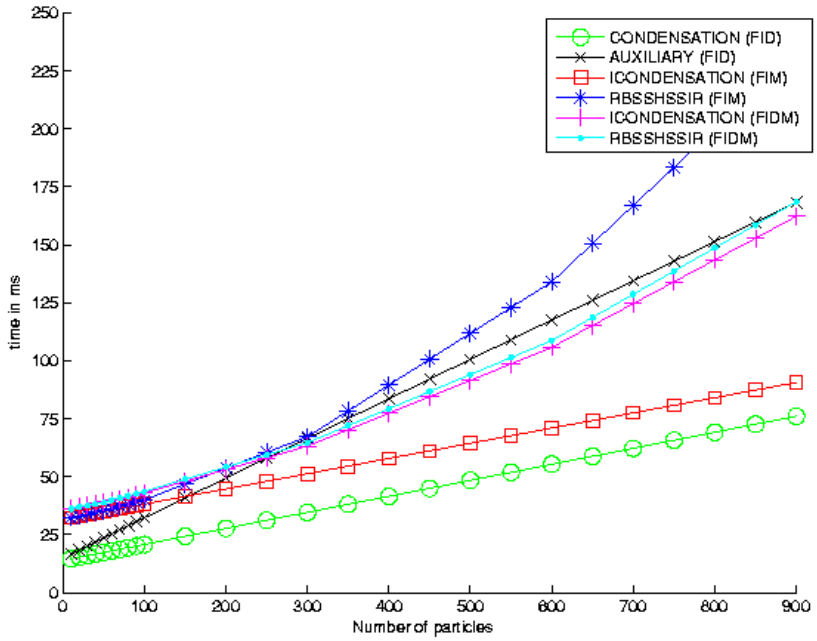
Failure ratio



TOP

Time consumption *vs* particles number:

Time consumption



---

2. Mean range tracking

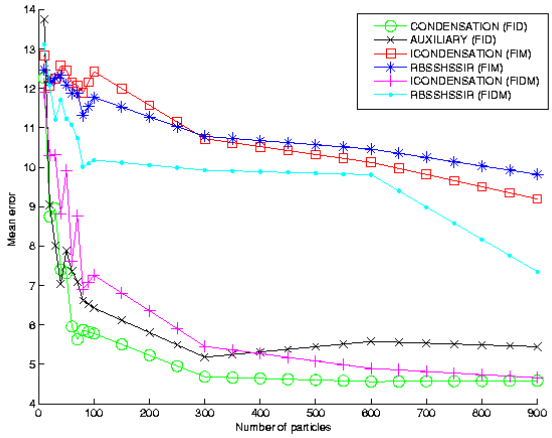
This modality merges shape and color distribution cues. An evaluation of several tracking strategies on representative sequences of the mean range tracking modality is presented below.

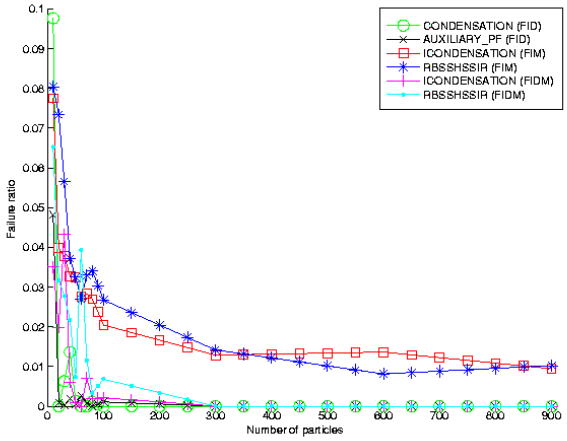
TOP

Nominal conditions

Video 6 : DMIF strategy in nominal conditions

Mean error **Failure ratio**





TOP

Illumination changes

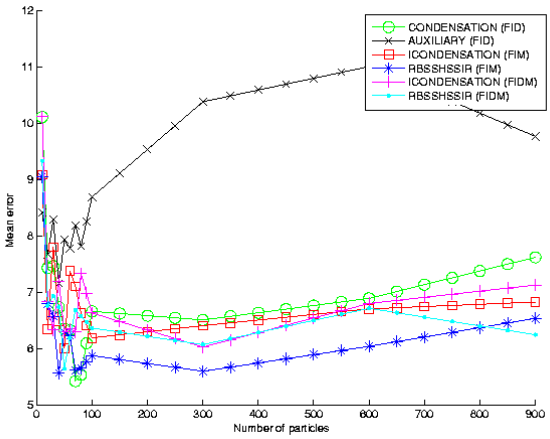
Video 7 : DMIF strategy in illumination changes

TOP

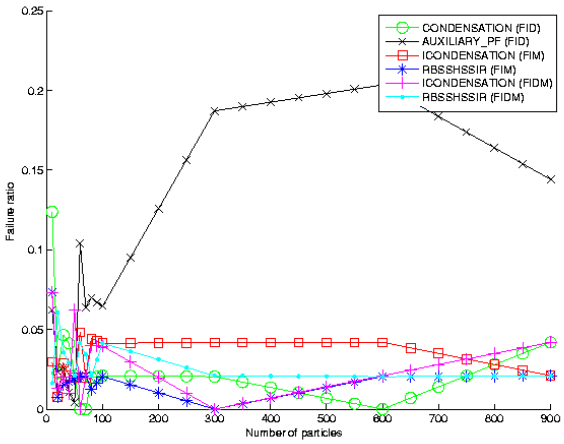
Appearance changes

Video 8 : DMIF strategy in appearance changes

Mean error



Failure ratio

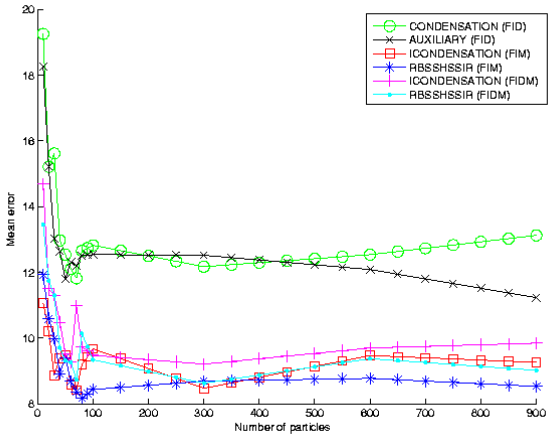


TOP

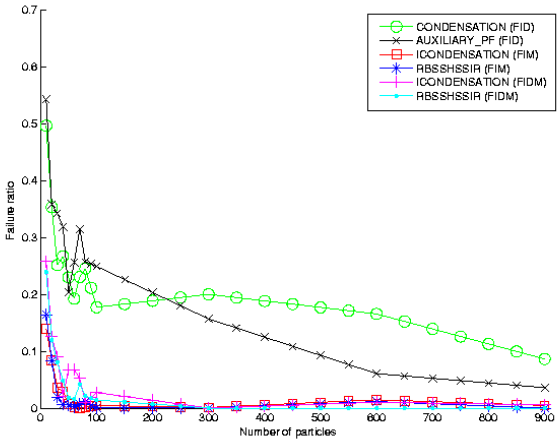
Jump in the target dynamics

Video 9 : DIF strategy **Video 10 : MIF strategy** **Video 11 : DMIF strategy**

Mean error



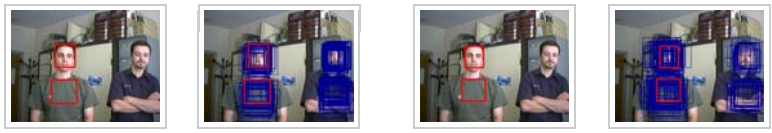
Failure ratio



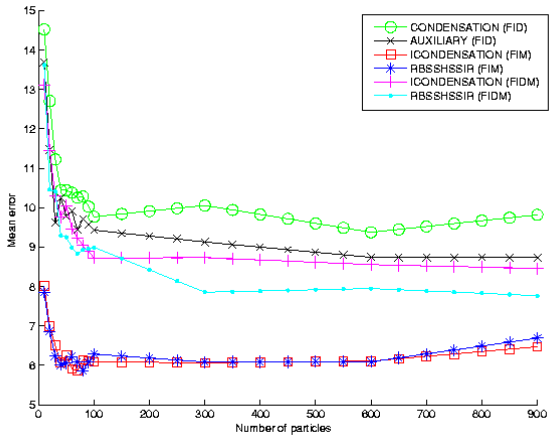
TOP

Two people without occultation:

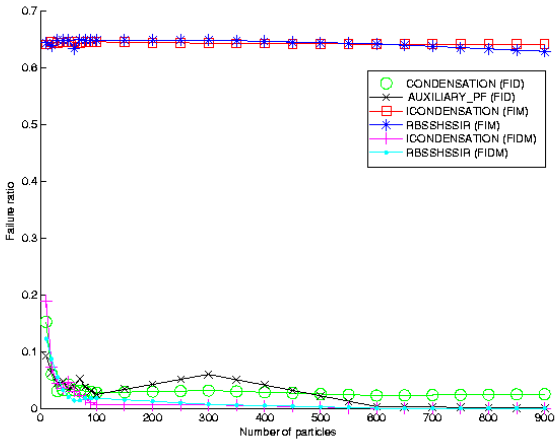
Video 12 : MIF strategy **Video 13 : DMIF strategy**



Mean error



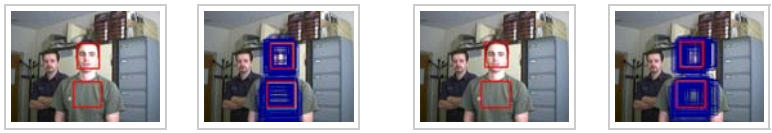
Failure ratio



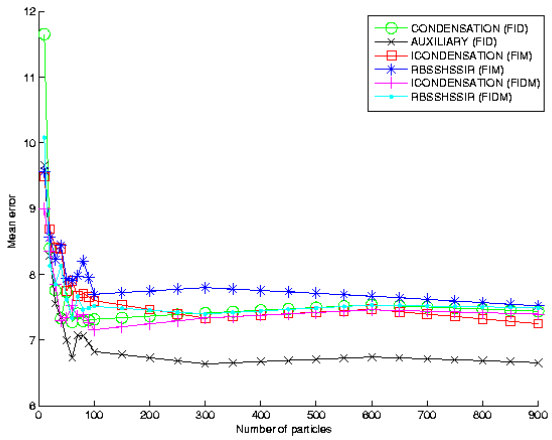
TOP

Two people occulting each other

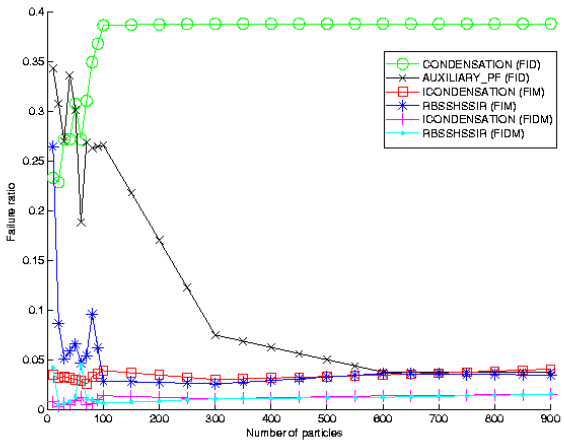
Video 14 : DIF strategy **Video 15 : DMIF strategy**



Mean error



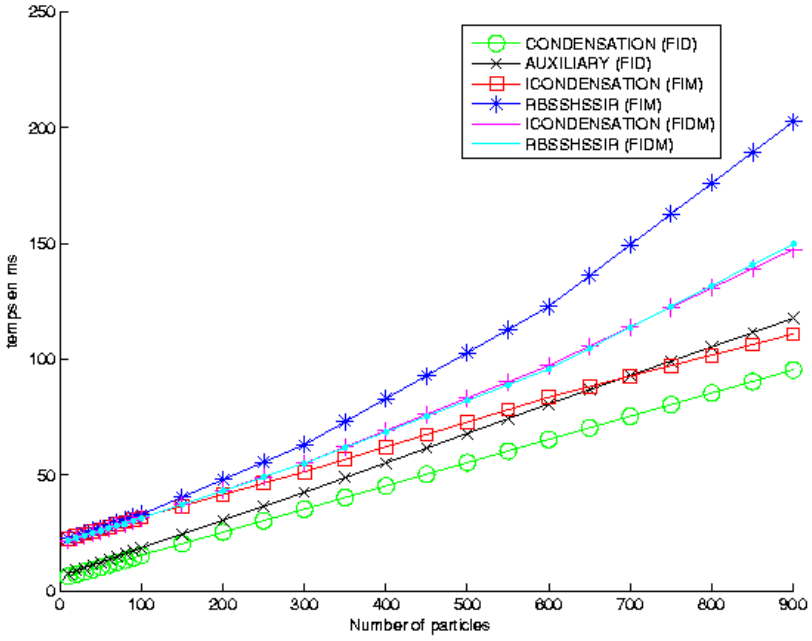
Failure ratio



TOP

Time consumption *vs* particles number:

Time consuming



3. Long range monitoring

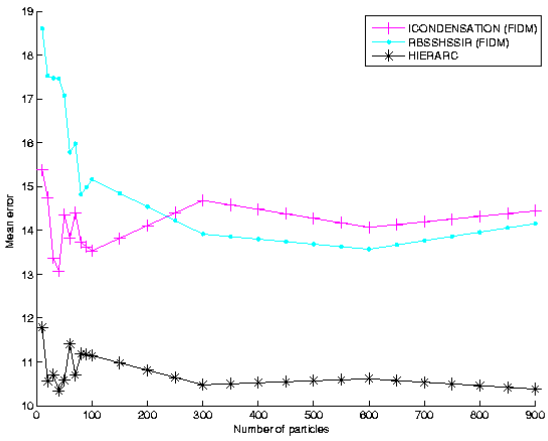
This modality merges motion and color distributions. An evaluation of various particle filtering strategies on representative sequences of the long range tracking modality is shown below.

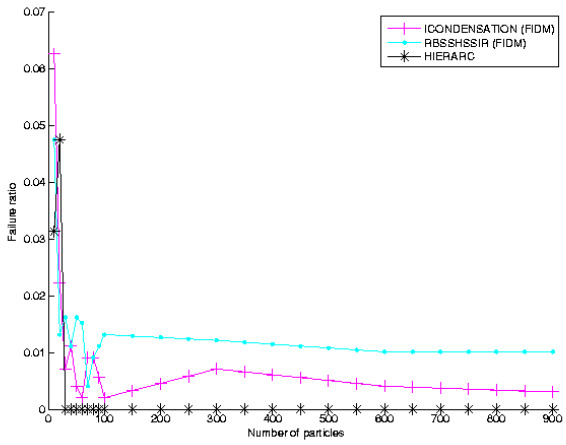
Nominal conditions

Video 16 : HIERARC strategy **Video 17 : DMIF strategy**




Mean error **Failure ratio**



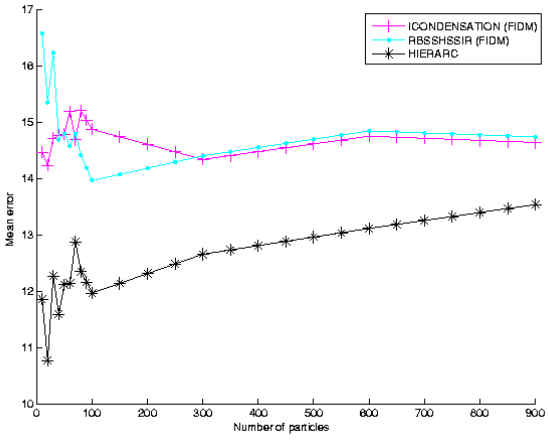


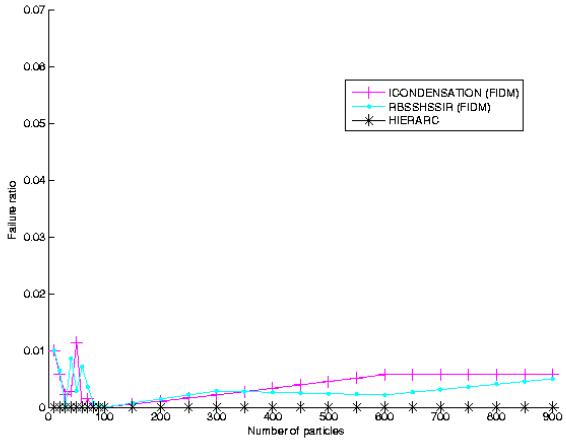
TOP

Sporadic pauses in the target motion

Video 18 : DMIF strategy


Mean error **Failure ratio**





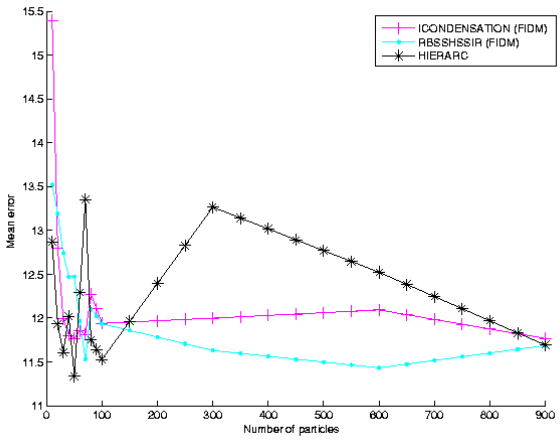
TOP

Full occultation by a static object

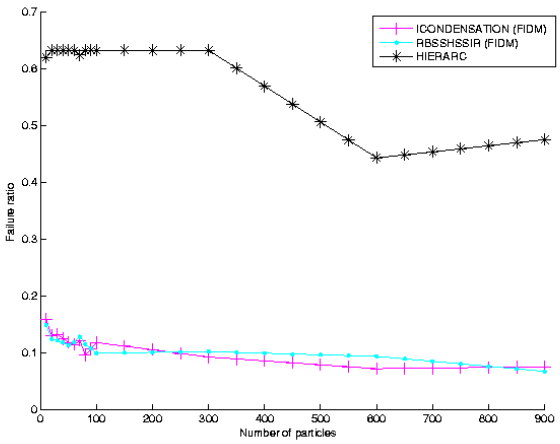
Video 19 : HIERARC strategy

Video 20 : DMIF strategy

Mean error



Failure ratio

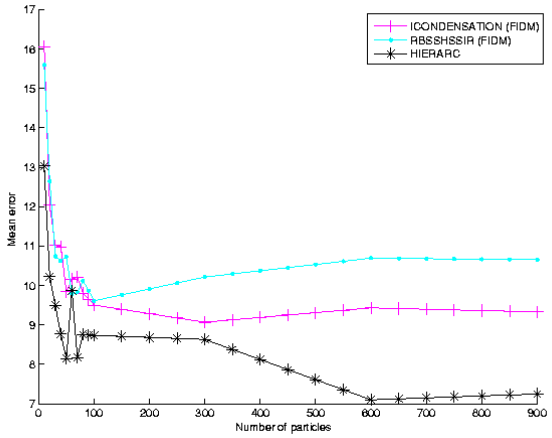


TOP

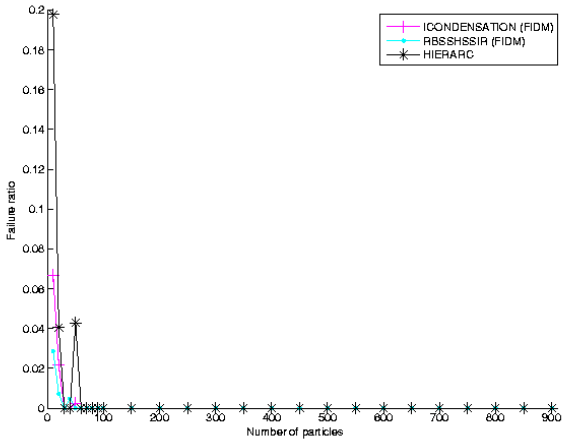
Two people occulting each other

Video 21 : DMIF strategy

Mean error



Failure ratio

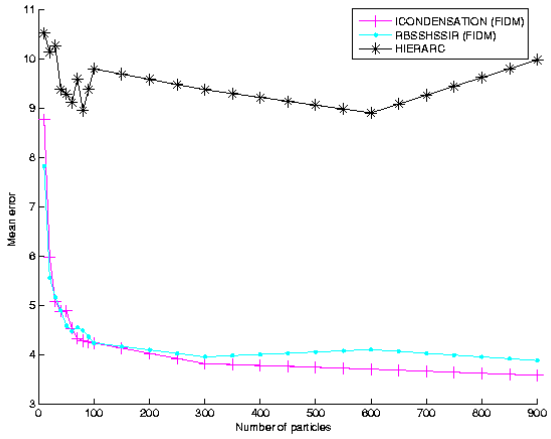


Occultation of motionless target

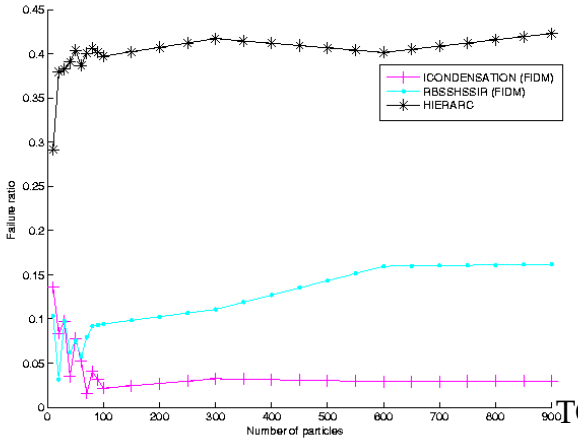
Video 22 : HIERARC strategy **Video 23 : DMIF trategy**




Mean error



Failure ratio



TOP

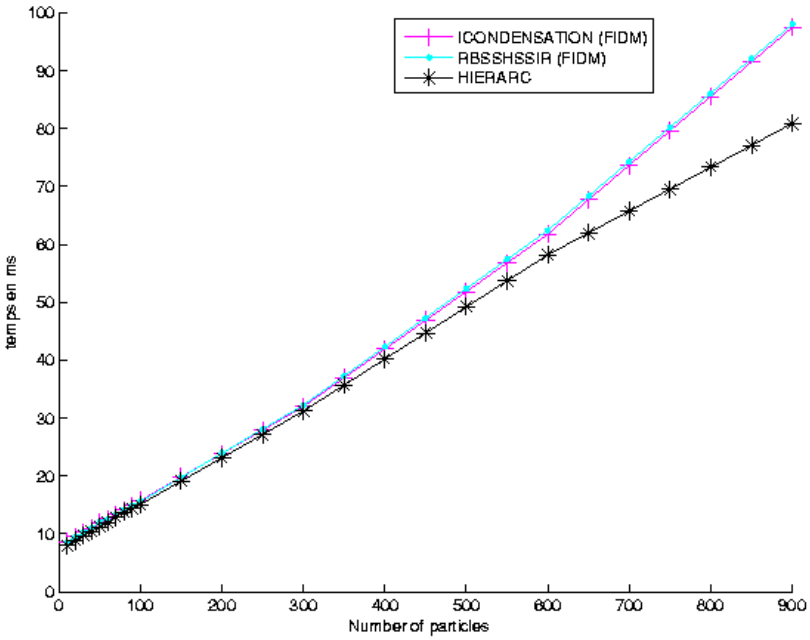
Group of people

Video 24 : DMIF strategy


TOP

Time consumption *vs* particles number:

Time consuming



Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model

Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann
Industrial Applications of Informatics and Microsystems
Institute of Computer Science and Engineering
University of Karlsruhe, Germany
Email: {knoop,vacek,dillmann}@ira.uka.de

***Abstract*— This paper proposes a tracking system called *VooDoo* for 3d tracking of human body movements based on a 3d body model and the *Iterative Closest Point* (ICP) algorithm. The proposed approach is able to incorporate raw data from different input sensors, as well as results from feature trackers in 2d or 3d. All input data is processed within the same model fitting step by modeling all input measurements in 3d model space. The system has been implemented and runs in realtime at appr. 10-14 Hz. Experiments with complex human movements exhibit the characteristics and advantages of the proposed approach.**

I. Introduction

Robots that are meant to cooperate closely with humans, and especially with untrained persons which are not familiar with the domain of robotics, need a deep understanding of the intentions, activities, actions and movements of their interaction partner.

This is on the one hand due to the fact that the robot needs the ability to predict the global plan as well as single movements of the human in order to plan its own actions and movements in an efficient way with respect to the overall goal. Even if parts of the goal can be explicitly communicated between human and robot, there are in most cases several ways to reach a given goal, especially in a cooperation context. Thus, not only motion prediction, but also activity recognition is an indispensable feature for such a robot.

On the other hand, a shared workspace between robot and human puts up high safety demands. This includes not only collision detection, but also haptic interaction and shared object and tool manipulation. Therefore, observation and prediction of the human's movements is badly needed in a robot system that is designed to work together with humans.

Many tracking systems for humans have been proposed in literature, some of which are discussed in sec. II. Most of these are designed for one special input sensor, and all internal models are based on this assumption.

This paper introduces a 3d body model based tracking system called *VooDoo*, and especially proposes a new approach for fusion of different input sensors and cues for tracking. This approach is able to incorporate tracking information from 3d sensors like *Time-of-Flight*-cameras (ToF) or stereo reconstruction together with cues from 2d based trackers like a monocular camera. The system is designed to work only with sensors on-board the robot.

The system is able to track a person in realtime at about 10-14 Hz in 3d. Results are shown with different input sensors.

II. State of the Art

For observation and tracking of human movements, many different sensors and models have been used. This includes invasive sensors like magnetic field trackers (see [1], [2]) that are fixed to the human body. Within the context of human robot interaction in every-day life, this approach is not feasible; non-invasive tracking approaches must be applied. Most of these are based on vision systems, or on multi-sensor fusion (see [3]). Systems which rely on distributed sensors (see [4]) are not practicable in the given domain; the tracking system must be able to rely only on sensors mounted on the robot.

Tracking of humans and human body parts using vision is investigated by a lot of research groups and several surveys exist (see [5], [6], [7], [8]). Hence, there is a big variety of methods ranging from simple 2d approaches such as skin color segmentation (e.g. [9]) or background subtraction techniques (e.g. [10]) up to complex reconstructions of the human body pose. [11] shows how to learn the appearance of a human using texture and color.

Sidenbladh [12] used a particle filter to estimate the 3d pose in monocular images. Each particle represents a specific configuration of the pose which is projected into the image and compared with the extracted features. [13] use a *shape-from-silhouette* approach to estimate the human's pose. A similar particle filtering approach is used in [14]. The whole body is tracked based on edge detection, with only one camera. The input video stream is captured with 60 Hz, which implies only small changes of the configuration between two consecutive frames. As it is a 2d approach, ambiguities of the 3d posture can hardly be resolved.

An ICP-based approached for pose estimation is shown in [15]. The authors use cylinders to model each body part. In [16] the same authors show how they model joint constraints for their tracking process. However, it the effect of the ICP is partially removed when the constraints are enforced. Nevertheless, parts of the work described in this paper are based on the work of Demirdjian et al.

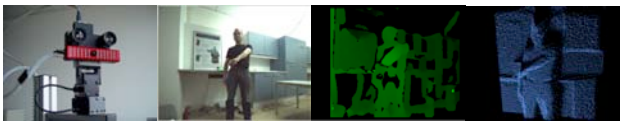


Fig. 1. Sensor head (left), 2d image (middle left), disparity image (middle right), 3d image (right)

III. Used Framework

This section describes the framework which is used for the presented work: Used sensors, the ICP algorithm which forms the basis, the articulated 3d human model and the joint model within the body model.

A. Sensor Data

In the described framework, two different sensors are used to demonstrate the capabilities of the algorithm: A time-of-flight (ToF) camera and a standard stereo camera head with depth data reconstruction generate 3d point clouds, and the color information of the camera is used to track face and hands with a simple skin color model in 2d.

The *Swissranger* ToF camera uses a resolution of 160×124 pixels. The output consists of a dense depth image and an intensity image. The depth range is configured to $0.5\ m \leq$ *range* $\leq 7.5\ m$, the accuracy lies within a few centimeters. Intensity data is not used within the current context, as the intensity image has very low resolution and high noise due to the sensor concept.

The stereo camera (*mega-d* from *videre design*) is used at a resolution of 320×240. The disparity image is computed based on a calibration obtained offline.

The sensors and the raw data can be seen in fig. 1.

B. Iterative Closest Point Algorithm

This section gives a short introduction to the *Iterative Closest Point (ICP)* algorithm. The goal of the ICP is to match two indexed sets of the same points which are given in different coordinate systems and calculate the translation $\vec{t}$ and rotation $\mathbf{R}$ that transform the first coordinate system into the second. For person tracking, the first set corresponds to the data points of the sensor and the second set corresponds to points on the surface of a rigid body. Following [17], the first set is denoted $P = \{\vec{p_i}\}$, the second one $X = \{\vec{x_i}\}$. Both sets have the same size with $N_x = N_p = N$ and each point $\vec{p_i}$ corresponds to point $\vec{x_i}$.

Because the sensor data is always corrupted with noise, no exact solution exists. Instead, the problem is transformed into the minimization of a sum of squared distances:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}) + \vec{t} - \vec{p_i}||^2 \qquad (1)$$

For a complete description of how to compute the optimal translation and rotation, see [18].

The sensor data consists of a list of data points, which has to be matched to a geometrical description of the body. To retrieve the ordered list of point pairs needed for the ICP, the correspondences between data and model have to be constructed.

This is done by calculating for each data point $\vec{p_i}$ the geometrically *closest point* on the model giving $\vec{x_i}$. In the second step the optimal translation and rotation can be estimated and applied to the model. This process is then repeated until the absolute value of the transformation is below some threshold. The *Iterative Closest Point* steps are:

1) For the given model and the data points calculate the closest points giving CP_0
2) Calculate the sum of squared distances between data points and model points giving $d_0(M, CP_0)$
3) Estimate rotation and translation and apply to the model
4) Calculate new set of closest point with the new position of the model giving CP_i
5) Calculate the sum of squared distances between data points and model points giving $d_i(M, CP_i)$
6) If $d_{i-1}(M, CP_{i-1}) - d_i(M, CP_i) < \epsilon$ the iteration stops, otherwise go to step 3.

Note that computation of *closest point relations* is by far the most time consuming step in the ICP process, since it includes a set of geometric calculations for each data point in the point cloud.

C. Human Body Model

For the tracking system a 3d body model is used. Each body part is represented with a *degenerated cylinder*.The top and the bottom of each cylinder is described by an ellipse. The ellipses are not rotated to each other and the planes are parallel.

The overall body model is built in a tree-like hierarchy starting with the torso as root body part. Each child is described with a degenerated cylinder and the corresponding transformation from its parent. Up to now the body model consists of ten body parts (torso, head, two for each arm and two for each leg) which is depicted on the left of fig. 2. It should be mentioned that this body model is not necessarily restricted to humans, and also other bodies can be modeled easily.

If the fusion algorithm also incorporates data from feature trackers (like some vision based algorithms, or magnetic field trackers that are fixed on the human body), it is required to identify certain feature points on the human body. This is done following the *H−Anim Specification* (see [19]).

D. Joint Model

The joint model is based on the concept of introducing elastic bands into the body model. These elastic bands represent the joint constraints. For the ICP algorithm, these elastic bands can be modeled as artificial correspondences and will thus be considered automatically in each computation step (see sec. IV-B.6).

For each junction of model parts, a set of elastic bands is defined (see fig. 2). These relations set up corresponding points on both model parts. The corresponding points can then be used within the model fitting process to adjust the model

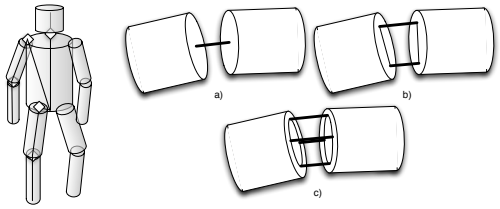


Fig. 2. Different joint type models. Universal Joint 3 DoF a), Hinge Joint 1 DoF + 2 restricted DoF b), Elliptic Joint 3 restricted DoF c)

configuration according to all sensor data input and to the defined constraints.

This approach allows for modeling of different joint types. Within the described tracking system, three types are used:

- **Universal Joints** have 3 full degrees of freedom (e.g. human shoulder).
  Universal joints are modeled by one point-to-point correspondence (one elastic band), see fig. 2 a).
- **Hinge Joints** have one real degree of freedom, the others being almost fixed (e.g. elbow or knee).
  Hinge joints are modeled by a set of correspondences which are distributed along a straight line, see fig. 2 b).
- **Elliptic Joints** have all degrees of freedom highly restricted. An example on the human body is the neck: Motion is possible in all 3 degrees of freedom, but very limited in range.
  Elliptic joints are modeled by a set of correspondences distributed along an ellipse, see fig. 2 c).

For details of the joint model, see also [20].

IV. Sensor Fusion Algorithm for Tracking

The goal of the *VooDoo* tracking system is to track the posture of a human body in 3d by matching the internal 3d body model with the current input sensor data. Thus, the tracking system offers three interfaces: sensor data stream (input), parameter configuration (input), and current posture estimation (output). All sensor data formats that can be exploited are described in section IV-A. The configuration values we have identified will be described in sec. IV-B along with the processing steps.

The current posture estimation output is given with respect to the hierarchical body model defined in sec. III-C. In each time step, the whole body model is provided. This allows for changes not only in the body pose (joint angle space), but also for changes in the model itself (configuration and parameters of the body model). This may concern scaling of the model for different persons with varying body heights, or even addition and deletion of body parts in case of changing tracking targets or other effects. This can be useful e.g. if the tracked person is holding and handling a big object, which then can be added easily to the tracked configuration.

The *VooDoo* tracking algorithm is depicted in fig. 3. The next section describes possible input data, while sec. IV-B depicts the processing steps within the tracking loop.

A. Input data

The proposed tracking algorithm is able to include, process and fuse different kinds of sensor data (see also fig. 3):

- *Free 3d points* from ToF-sensors or from pure stereo depth images. The system has to decide whether to use these points as measurements of the tracked model. For a point that is not discarded, the corresponding point on the model surface is computed.
- *3d points on the human body* that are e.g. generated by a stereo vision system that tracks a person in image space and generates the corresponding 3d points by stereo reconstruction.
- *3d points assigned to a single body part* may also be generated by a stereo vision system tracking special body parts like the face or the hands.
- *3d point-to-point relations* are 3d points that can be assigned to a given point on the tracked human body. Thus, tracking of special features or points (e.g. with markers, or magnetic field trackers attached to the human body) can be integrated.
- *2d point-to-line relations* can e.g. be derived from a 2d image space based tracker. The pixel in the image plane together with the focal point define a ray in 3d, which corresponds to the point on the human body that has been detected in the image.

This data can originate from any sensor that gives data in the described format. Obviously, all input data has to be transformed into the tracker coordinate system before it is used within the system.

B. Processing

For the ICP matching algorithm, a list of corresponding point pairs has to be set up for each limb (see also sec. III-B). Therefore, all “free” 3d points have to be analyzed in order to decide whether they correspond to points on the tracked model. Otherwise, they are discarded. Additionally, all given correspondences from other tracking procedures and the background knowledge on joint constraints have to be added to the correspondences list. Then, the optimal resulting model configuration has to be computed. These steps are performed iteratively until an optimum of the configuration is reached.

Before the input data of one time step is processed, it is possible to adjust internal model parameters. This can be e.g. the model scale factor, or particular cylinder sizes. Even limbs can be added to or removed from the model.

The tracking algorithm and the sensor fusion approach are now described step by step.

1) Prefiltering free 3d points: The whole point cloud of free 3d points from used depth sensors is processed in order to remove all points that are not contained within the bounding box of the body model (see fig. 3, step *BB Check whole body*). This is done on the assumption that the body configuration changes only locally between two time frames. A parameter defines an additional enlargement of the bounding box prior to this filtering step.

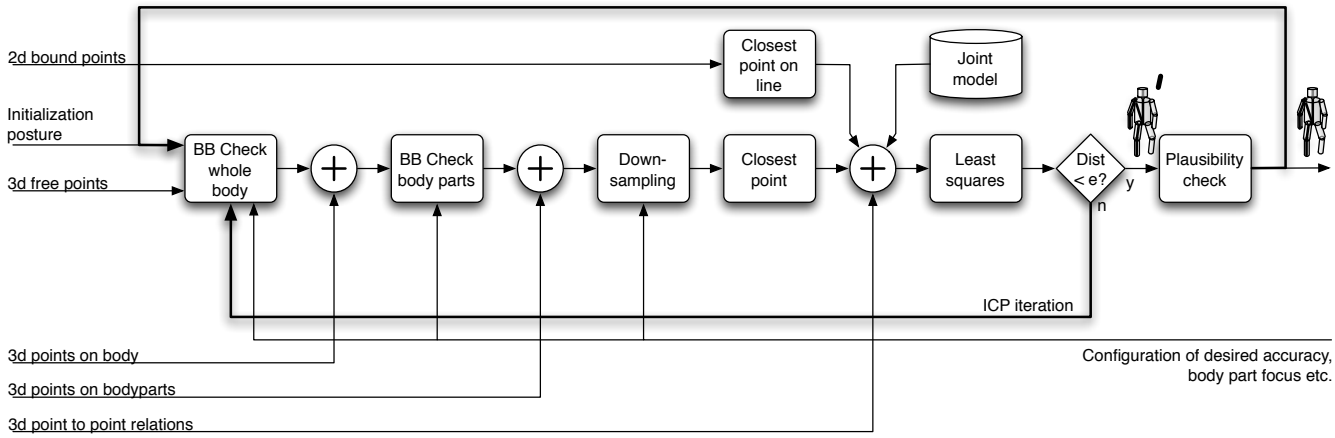


Fig. 3. The complete VooDoo algorithm ("BB" = Bounding Box)

The resulting point list is concatenated with any sensor data input that has already assigned its measured 3d points with the tracked body (see sec. IV-A). This results in a list of 3d points which are close to the body model and thus are candidates for measurements of the tracked body.

2) Assigning points to limb models: The point list is now processed in order to assign measured points to dedicated limb models based on the bounding box of each limb model (see fig. 3, step *BB Check body parts*). Again, the bounding boxes can be enlarged by a parameter to take the maximum possible displacement into account. Points that do not fall in any bounding box are again removed. Several behaviors can be selected for points that belong to more than one bounding box (overlap): These points are either shared between limb models, exclusively assigned to one limb or shared only in case of adjacent limbs. This last method avoids collisions between limbs that are not directly connected.

The resulting point list can be joined with any sensor data input that has already assigned its measured 3d points with dedicated limbs of the tracked body (see sec. IV-A). The resulting point list contains candidates for measurements of each limb.

3) Point Number reduction: The resulting point list can be downsampled before the calculation of the closest points to reduce the overall number of points (see fig. 3, step *Downsampling*). This step is controlled by three parameters: the sampling factor, and minimum and maximum number of points per limb. Thus, it is possible to reduce the number of points for limbs with many measurements, but maintain all points for limbs which have been measured with only a few points.

4) Closest point computation: The closest point calculation is the most time-consuming step in the whole loop. For each remaining data point, the corresponding model point on the assigned limb model has to be computed for the ICP matching step (see fig. 3, step *Closest Point*). This involves several geometric operations. Depending on the resulting distance between data and model point, all points within a given maximum distance are kept and the correspondence pair is stored in the output list. All other points are deleted.

3d point-to-point relations from input data (see sec. IV-A) can now be added to the resulting list, which holds now corresponding point pairs between data set and model.

5) Addition of 2d measurements: Each 2d measure (e.g. tracked features in 2d image plane of a camera) of a feature on the human body defines a ray in 3d which contains the tracked feature. This fact is used to add the 2d tracking information to the 3d point correspondences (see fig. 3, step *Closest point on line*): For each reference point on the body model, the closest point on the straight line is computed and added to the list.

6) Joint model integration: The joint model for each junction is added as artificial point correspondences for each limb, depending on the limb type (see fig. 3, step *Joint model*). According to sec. III-D, the correspondences can be interpreted as elastic bands which apply dedicated forces to the limbs to maintain the model constraints. Thus, artificial correspondences will keep up the joint constraints in the fitting step.

7) Model fitting: When the complete list of corresponding point pairs has been set up, the optimal transformation between model and data point set can be computed according to sec. III-B (fig. 3, step *Least squares*). The transformation is computed seperately for each limb.

When all transformations have been computed, they can be applied to the model. The quality measure defined in sec. III-B is used for the fitting. Steps IV-B.1 to IV-B.7 are repeated until the quality measure is below a given threshold or a maximum number of steps have been performed.

C. Sensor model

Each used data source has its own stochastic parameters which have to be taken into account. The described approach offers a very simple method for this: each input date is weighted with a measure that describes its accuracy. The ICP algorithm then incorporates these weights in the model fitting

step. Thus it is possible to weight a 2d face tracker much higher than a single 3d point from a ToF camera.

It is important to note that an increased weight for a single point does not affect the time needed for the computation.

V. Experiments and Results

The described tracking procedure has been implemented and tested with the sensors described in sec. III-A. The tracking runs online at a framerate of appr. 10-14 Hz on a Pentium 4 with 3.2 GHz.

Different test series have been performed to evaluate the *VooDoo* system: First, the same data sequences have been processed using different input sensor configurations to test the fusion, and second, a set of 100 sequences has been recorded and processed. The tracking result has then been evaluated manually for consistency with the recorded body movements to evaluate the overall system performance.

Fig. 4 shows example images from a sequence of 15 seconds containing a "bow" and a "wave" movement. The first row shows the scene image, which has been also used for segmentation of face and hands. The second and third row contain the tracking result with 3d data only (row 2) and 2d data only (row 3), where the 3d data has been acquired with the ToF camera and the 2d data is derived from skin color segmentation in one image of the stereo camera. The rays in 3d defined by the skin color features can be seen here. Row 4 shows the tracking result with both inputs used.

For the shown results, the following weights for the input data have been used: 3d data points $w = 1.0$, face tracker $w = 30.0$, hand tracker $w = 20.0$.

Different conclusions can be drawn from the results:

- Huge movements are easily detected by the 3d data based tracking: The "bow" movement is tracked quite well. On the other hand, fast movements with the extremities may cause failures when only 3d data is used, as with the "wave" movement.
- Tracking only with a 2d feature tracker works quite well for the tracked body parts. Nevertheless, the body configuration can not be determined only from 2d features (see frame 81). To do this, a lot more background information on the human body would be needed.
- Fusion of both input sensors in 3d shows very good results: Huge body movements as well as fine and fast movements of the extremities can be recognized, and the algorithm is able to reliably track the body configuration.

The second evaluation step consisted in recording a set of 100 sequences which contained ten different movements from several persons: e.g. *point somewhere*, *walk*, *wave*, *shake hands* with somebody, *bow* or *clap*. The tracking result has then been evaluated and classified manually into one of three classes: (0) *Tracking lost* somewhere within the sequence, (1) *acceptable deviations* like a temporally lost (but recovered) forearm within a walking sequence, and (2) *good congruence* between original and resulting model movements. The evaluation result is depicted in tab. I, the average result is $\oslash = 1.58$.

TABLE I
Evaluation result with 100 sequences

Tracking result	0	1	2
# of sequences	5	32	63

VI. Discussion

The proposed tracking approach does not include any background knowledge apart from kinematic constraints, i.e. no assumptions like "the torso stands always upright" are made. This implies on the one hand that all possible configurations can be recognized; on the other hand, the tracking can only succeed if the input data contains all necessary information to determine the human posture, and no tracking hypothesis can be generated for temporarily invisible body parts or ambiguous configurations.

The current framerate is appr. 10-14 Hz. The computation time depends on several factors: It scales linearly with the number of measured 3d points on the model; background points are removed in an early stage and do not distinctly influence framerate. It also depends on the number of ICP steps performed in each frame, which is appr. 3-15, depending on the desired accuracy and the speed of the movement.

Sec. V has shown that tracking based only on the measurements of the ToF camera is not sufficient. Especially movements along the main axis of the body (e.g. sitting down) can hardly be detected, which substantiates again the use of different data inputs for a fusion algorithm.

VII. Conclusion

This paper has proposed a new way for fusion of different input cues for tracking of a human body. The proposed algorithm is able to process 3d as well as 2d input data from different sensors like ToF-cameras, stereo or monocular images. It is based on a 3d body model which consists of a set of degenerated cylinders, which are connected by an *elastic bands* joint model. The proposed approach runs in realtime and is able to track complex movements like walking or bowing. It even recognizes postures with the arms outstretched directly towards the sensor.

The described way of adding 2d measurements to a 3d matching process is one of the main innovations. The idea of adding artificial point correspondences from non-3d sensors or background knowledge to the 3d matching process can even be exploited further: Future works will investigate methods to include valid ranges for joints via addition of artificial correspondences. Other unsolved issues are the initialization process, or the computation of an optimal scale factor for the model to incorporate the ability to track persons of different height without manually resizing the model.

References

[1] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann, "Observation in programming by demonstration: Training and execution environment," in *Proceedings of Third IEEE International Conference on Humanoid Robots, October 2003, Karlsruhe*, Karlsruhe and Munich, Germany, 2003.

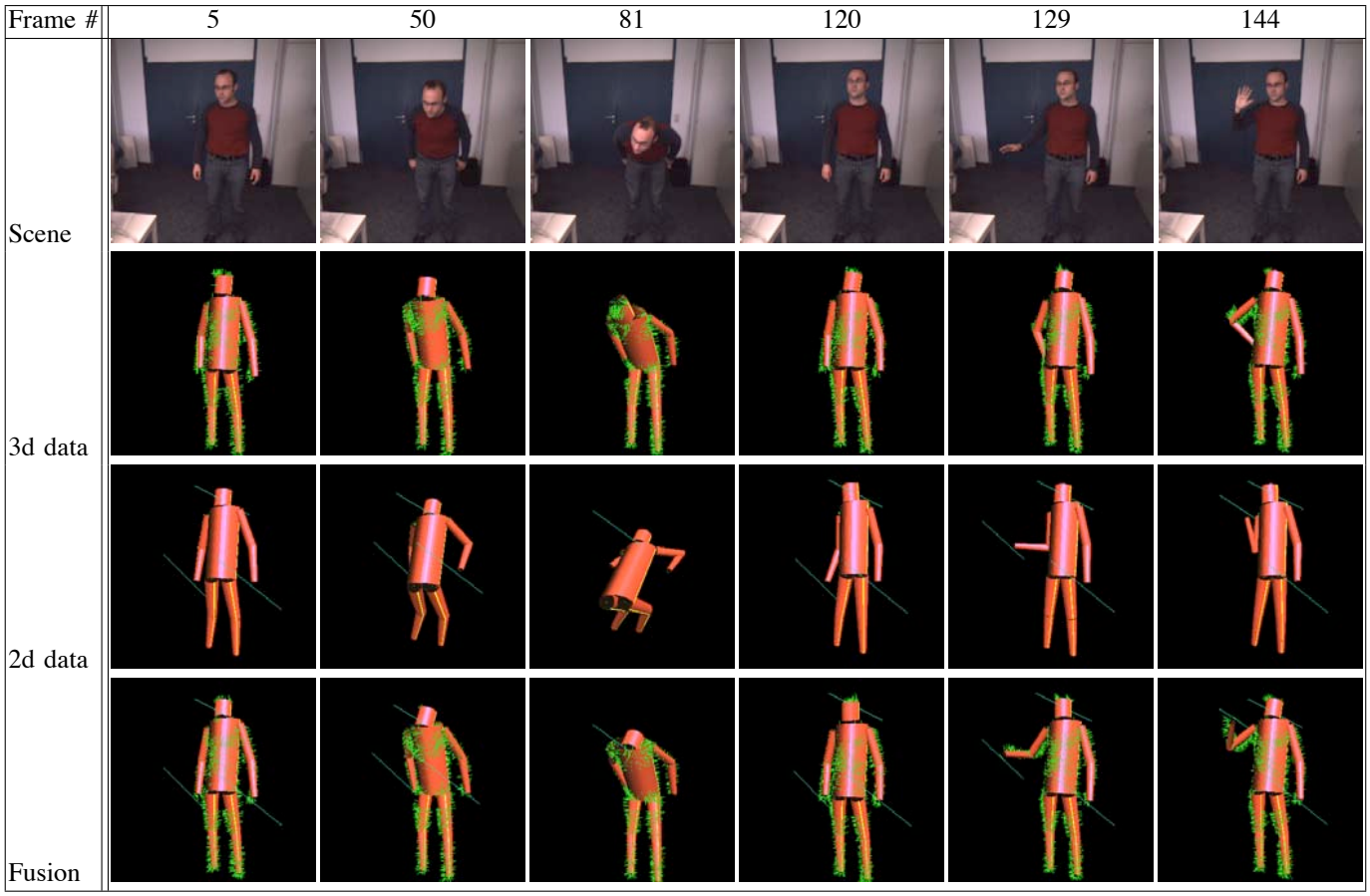


Fig. 4. Experiments with different sensor inputs, taken from a sequence containing a "bow" and a "wave" movement. The frame number is displayed on the top. The used 2d and 3d correspondences have been added to the resulting model images.

[2] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm," in *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

[3] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer, "Multi-modal anchoring for human-robot-interaction," *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, vol. 43, no. 2–3, pp. 133–147, 2003.

[4] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, USA, 2000, pp. 2126–2133.

[5] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, pp. 428–440, 1999.

[6] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[7] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231–268, 2001.

[8] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[9] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, and G. Sagerer, "Improving adaptive skin color segmentation by incorporating results from face detection," in *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN)*. Berlin, Germany: IEEE, September 2002, pp. 337–343.

[10] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 257–267, 2001.

[11] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *Computer Vision and Pattern Recognition*, vol. 2, 18-20 June, 2003, pp. II–467–II–474.

[12] H. Sidenbladh, "Probabilistic tracking and reconstruction of 3d human motion in monocular video sequences," Ph.D. dissertation, KTH, Stockholm, Sweden, 2001.

[13] G. K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *Computer Vision and Pattern Recognition*, 2003.

[14] P. Azad, A. Ude, R. Dillmann, and G. Cheng, "A full body human motion capture system using particle filtering and on-the-fly edge detection," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*. Santa Monica, USA: IEEE Institute of Electrical and Electronics Engineers, 2004.

[15] D. Demirdjian and T. Darrell, "3-d articulated pose tracking to untethered diectic references," in *Multimodel Interfaces*, 2002, pp. 267–272.

[16] D. Demirdjian, "Enforcing constraints for human body tracking," in *2003 Conference on Computer Vision and Pattern Recognition Workshop Vol. 9*, Madison, Wisconsin, USA, 2003, pp. 102–109.

[17] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.

[18] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Optical Society of America Journal A*, vol. 4, pp. 629–642, Apr. 1987.

[19] Humanoid Animation Working Group, "Information technology — Computer graphics and image processing — Humanoid animation (H-Anim), Annex B," ISO/IEC FCD 19774 - Humanoid Animation," Specification, 2003.

[20] S. Knoop, S. Vacek and R. Dillmann, "Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP," in *Proceedings of the International Conference on Humanoid Robots (Humanoids 2005)*. Tsukuba, Japan: IEEE-RAS, 2005.

Human-like Person Tracking with an Anthropomorphic Robot

Abstract— **A very important aspect in developing robots capable of human-robot interaction (HRI) is natural, human-like communication. Besides a flexible dialog system and speech understanding an anthropomorphic appearance shows many advantages in intuitive usage and understanding of a robot. As a consequence of our effort in creating an anthropomorphic appearance and to keep as close as possible to a human-human interaction, we decided to use human-like sensors, i.e., two cameras and two microphones only, not using a laser range finder or omnidirectional camera for tracking persons. Despite the challenge of a limited field of perception, we created a robust attention system for tracking and interacting with multiple persons simultaneously in real-time. Our approach is sufficient generic to work on robots with varying hardware, as long as stereo audio data and images of a video camera are available. Since the architecture is designed modular with a XML based data exchange we are able to extend the robot's abilities easily.**

I. Introduction

For many people the use of technology has become common in their daily lives. Examples range from DVD players, computers at workplace or for home entertainment, to refrigerators with integrated Internet connection for ordering new food automatically. The amount of technique in our everyday life is continuously growing. Keeping in mind the raising mean age of the society and considering the increasing complexity of technology, the introduction of service robots that will act as interfaces with technological appliances is a very promising approach. An example for an almost ready to sale user-interface robot is iCat [1] focusing the target not to adapt ourselves to technology but being able to communicate with technology in a human way.

One condition we call for a successful application of these robots is the ability to easily interact with them in a very intuitive manner. The user should be able to communicate with the robot by natural speech without reading an instruction manual. Both the understanding of the robot by its user and the understanding of the user by the robot is very important for the acceptance of a robot. For a better and more intuitive understanding of the answers of a robot a human-like outward appearance of the robot and the ability of expressing mimics and gestures by the robot will be very useful. Hence, in our scenario we use a humanoid robot to interact with a human (see Fig. 1), which is able to express mimics on its face and uses its arms and hands for deictic gestures. The interaction recently consists of greeting the robot and showing objects to it, which are lying on a table, by deictic gestures of a human advisor [2]. For future work we can extend this interaction easily taking advantage of the modular architecture we chose. For a successful communication a robust and continuous tracking of possible communication partners and an attention control for interacting with multiple persons is obligatory.

Fig. 1. We want to intuitively interact with XXXX in a human like manner.

Due to an anthropomorphic appearance and our intention to research human-like interactions, sensors comparable to human senses, such as video cameras representing eyes and microphones instead of ears must be used. This paper will present a modular system, that is capable to find, continously track and interact with communication partners in real-time without the use of wide area sensors like laser range finders or omnidirectional cameras. We are able to run this system both on the stationary humanoid robot described in this paper and on a mobile service robot [3].

The paper is organized as follows: At first we discuss related work on anthropomorphic head torso robots and their interaction capabilities in section II. Section III introduces our anthropomorphic robot XXXX and in section IV, our approaches in detecting people by faces and voices for the use with different robots are shown. Subsequently, the combination of the modules to a working memory-based person selection and tracking is outlined in section V and section VI describes experiments showing the robustness of our software. The paper concludes with a summary in section VII.

II. Related Work

There are several human-like or humanoid robots used for research in human-robot interaction. ROBITA [4] for example is a head torso robot that uses a time and person-dependent attention system to take part in a conversation with several human communication partners. The robot is able to detect the gazing direction and gestures of humans by video cameras integrated in its eyes. The direction of a speaker is determined by microphones. The face detection is based on skin color detection at the same height as the head of the robot. The gazing

direction of a communication partner's head is estimated by an *Eigenface*-based classifier. For gesture detection the positions of hands, shoulders, and elbows are used. ROBITA calculates the positions of different communication partners and predicts who is taking the next turn in a conversation and to whom it will pay attention.

Another robot, consisting of torso and head, is SIG which is capable of communicating with several persons, too. Like ROBITA it estimates the recent communication partner. Faces and voices are sensed by two video cameras and two microphones. The data is applied to Detection-Streams which cue records of the sensor data for a short time. Several Detection-Streams might be combined for the representation of a person. The interest, a person attracts, is based on the state of the Detection-Streams that correspond to this person. A preset behavior control is responsible for the decision taking process how to interact with new communication partners, e.g., friendly or hostile [5].

A robot with a very human-like appearance and human-like movements is Repliee-Q2 [6]. It is designed to look like an Asian woman and uses 42 actuators for movements from its waist up. It is capable of showing different facial expressions and uses its arms for gestures. But till now no attention control or dialog system for interacting with several people simultaneously has been published.

A less human-like looking robot is Alpha [7]. Alpha uses 21 degrees of freedom to control its body and 16 degrees of freedom for facial expressions. It is able to communicate with multiple person relying on video and sound data only. The video data is generated by two video cameras fixed in its eyes and used for face detection. Two microphones are used for sound localization that is tracking only the most dominant sound source.

However, a major problem in current robotics is coping with a very limited sensor field. Our work enables XXXX to operate with these handicaps. For people within the sight of the robot we are able to track multiple faces simultaneously. To detect people who are out of the field of view we track not only the most intense but multiple sound sources, subsequently verifying if a sound belongs to a human or is noise only. Furthermore, the system memorizes people, who got out of sight by the movement of the robot.

III. Robot Hardware

We present the humanoid robot XXXX that was developed by [8]. XXXX is able to move like a sitting human and corresponds to an adult person with the size of 75 cm from its waist upwards. The torso is mounted on a 65 cm high chair-like socket, which includes the power supply, two serial connections to a desktop computer, and a motor for rotations around its main axis. One interface is used for controlling head and neck actuators, while the second one is connected to all components below the neck. Without the socket the weight of the robot is about 35kg to keep robot and socket easy to transport. The torso of the robot consists of a metal frame with a transparent cover to protect the inner elements. Within the torso all necessary electronics for movement are integrated. All

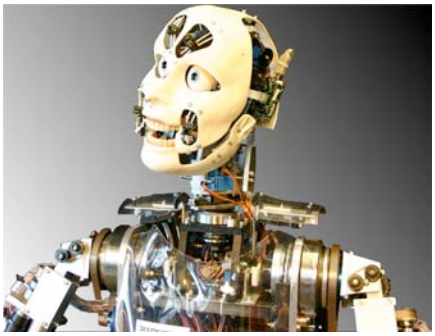


Fig. 2. The head of XXXX without skin

considered 41 actuators consisting of DC- and servo motors are used to control the robot hardware. To achieve human-like facial expressions ten degrees of freedom are used in its face to control jaw, mouth angles, eyes, eye brows and eye lids. The eyes are vertically aligned and horizontally steerable autonomously for object fixations. Each eye contains one FireWire color video camera with a resolution of 640x480 pixels. Besides facial expressions and eye movements the head can be turned, tilt to its side and slightly shifted forwards and backwards. The set of human-like motion capabilities is completed by two arms, mounted at the sides of the robot. Each arm can be moved like the human model with his joints. With the help of two five finger hands both deictic gestures and simple grips are realizable. The fingers of each hand are controllable autonomously and made of synthetic material to achieve minimal weight. Besides the neck two shoulder elements are added that can be lifted to simulate shrugging of the shoulders. For speech understanding and the detection of multiple speaker directions two microphones are used, one fixed on each shoulder element. This is a temporary solution. The microphones will be fixed at the ear positions as soon as an improved noise reduction for the head servos is available. By using different latex masks we can change the appearance of XXXX for different kinds of interaction experiments from a male youngster to an old woman. For extended experiments we have a second and smaller version of the robot with the appearance of a child.

IV. Detecting People

With the utilization of a video camera and two microphones we are able to detect and continously track multiple people in real-time with a robustness comparable to systems using wide field of perception sensors. To cope with different kinds of sensors the software is modular, consisting of autonomous running components communicating by the exchange of XML data with other subsystems, e.g., a dialog or an attention control. This software design enables us to run it on different robots. A previous version was running on our service robot, relying very strong to a laser range finder, which was necessary as well for detection and tracking persons as for the attention system. For the perception of XXXX we use two modules, one for face detection and one for the location of various speakers, which are described in detail in the following.

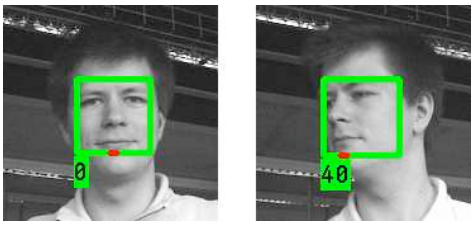


Fig. 3. Besides the detection of a face we classify the gazing direction in 20° steps to estimate the addressee in a conversation.

A. Face Detection

For face detection we use a method originally developed by Viola & Jones for object detection [9]. This approach divides the image in all different possible sub parts. Any part is classified as face or non face. Since many parts exist the classification process must be very efficient and fast. We use the idea of a classification pyramid [10] starting with very fast but weak classifiers to reject image parts that are certainly no faces. With increasing complexity of classifiers, the number of remaining image parts decreases. The training of the classifiers is based on the *AdaBoost* algorithm, combining the weak classifiers iteratively to more stronger ones until the desired level of quality is achieved.

Additionally to the detection of multiple faces in one image, we classify the horizontal gazing direction of faces as shown in Figure 3 by using four of the classifier pyramids described above, trained for faces rotated by 20°, 40°, 60° and 80°. For classifying left and right turned faces the image is mirrored at its vertical axis and the same four classifiers are applied again. We evaluate the gazing direction for activating or deactivating the speech processing, since the robot should not react to people talking to each other in front of the robot, but only to communication partners facing the robot. Subsequent to the face detection a face identification is applied to the detected image region using the Eigenface method to compare the detected face with a set of trained faces. For each detected face the size, center coordinates, horizontal rotation, and results of the face identification are provided at a real-time capable frequency of about 7*Hz* on an Athlon64 2GHz desktop PC with 1GB RAM. Due to the very limited field of view for tracking people by face detection we needed an alternative to find people who are out of the robot's sight.

B. Voice Detection

We use the SPeaker LOCalization (*SPLOC*) both for detecting possible communication partners outside the field of view and for determining whether a person found by face detection is speaking. The program continously records the audio data by the two microphones to estimate the relative direction of one or more sound sources in front of the robot. Therefore, the direction of sound towards the microphones is considered (see figure 4). Dependent on the position of a sound source (Fig. 4 (5))in front of the robot the run time difference Δ_t results from the run times t_r and t_l to the right and left microphone. SPLOC compares the recorded audio signal of the left (Fig. 4 (1)) and the right (Fig. 4 (2)) microphone and uses the result of a *Cross Power Spectrum Phase* (*CSP*)

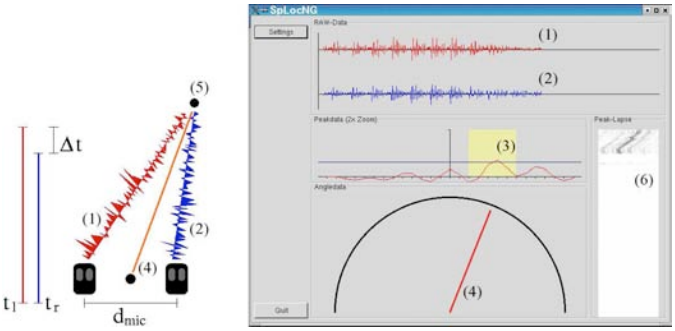


Fig. 4. Recording the left (1) and right (2) audio signals with microphones, the speaker localization (SPLOC) uses the result of a CSP (3) for calculating the difference Δ_t of the signal runtime t_l and t_r and estimates the direction (4) of a sound source (5) over time (6).

[11] (Fig. 4 (3)) to calculate the temporal shift between the signals. Taking the distance of the microphones d_{mic} and a minimum range of XXX cm to a sound source into account it is possible to estimate the direction (Fig. 4 (4)) of a signal in a two dimensional space. For the three dimensional space we need distance and height of a sound source detection. We assume the position of a sound source (a speakers mouth) at the height of 160 cm for an average adult. The standard distance is adjusted to 110 cm, as observed during interactions with naive users. If a face position is assignable to the sound source a more precise calculation is done, deciding whether the sound comes from the corresponding person or not. Running on an Intel P4 2.4GHz desktop PC with 1GB RAM we achieve a calculation frequency of about 8Hz, which is sufficient for a robust tracking as described in the next section.

V. Memory Based Person Tracking

For a continuous tracking of several persons in real-time we use the Anchoring approach developed by Coradeschi and Saffiotti [12]. Based on this we developed a variant that is capable of tracking people in combination with our attention system [13]. In a previous version no distinction was done between persons who did not generate percepts within the field of view and persons who were not detected because they were out of sight. We extended our system that a person might not be found because he might be out of the sensory field of perception. As a result we successfully avoided the use of a laser range finder like many other service robots do. To achieve a robust person tracking we had to handle two requirements. First, a person should not need to deal with starting at a certain position in the front of the robot. The robot ought to turn itself to a speaking person trying to get him into sight. Second, due to showing objects or interacting with another person the robot might turn and as a consequence the previous potential communication partner could get out of its sight. Instead of loosing him the robot will remember his position and return to it after the end of the interaction with the other person. It is not very difficult to add a behavior that makes a robot look at any noise in its surrounding that exceeds a certain threshold, but this idea is far too unspecific. At first we filter noise in SPLOC by a band-pass filter only accepting sound within the frequencies of human speech which

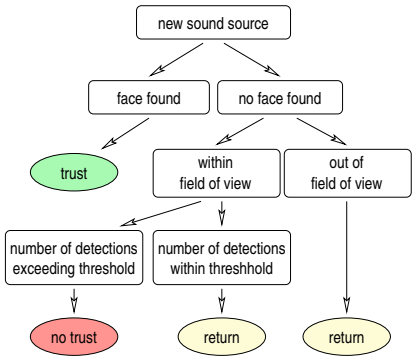


Fig. 5. Voice Validation. A new sound source is trusted if an according face is found. If the sound source is out of view the decision will be delayed until it gets into sight. Otherwise, a face must be found within a given number of detections or the sound source will not be trusted. Different colors correspond to different results.

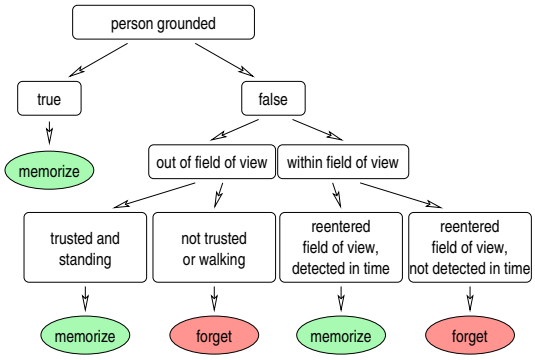
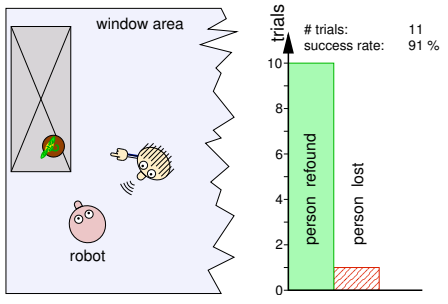


Fig. 6. Short time memory for persons. Besides grounded persons, the person memory will memorize people who were either trusted and not moving at the time they got out of sight or got into the field of view again but were not detected for a short time. Different colors correspond to different results.

is approximately between $100Hz$ and $4500Hz$. But the band-pass filter proved not sufficient for a reliable identification of human speech. As a radio or TV might attract the attention of the robot, too. We decided to follow the example humans give to us. If they encounter an unknown sound out of their field of view humans will possibly have a short look in the corresponding direction evaluating whether the reason for the sound arises their interest or not. If it does, they might change their attention to it, if not they will try to ignore it as long as the sound persists. This behavior is realized in the *Voice Validation* shown in Figure 5. Since we have no kind of sound classification, except the SPLOC, any sound will be of the same interest for XXXX and cause a short turn of its head to the corresponding direction looking for potential communication partners. If the face detection does not find a person there after an adjustable number of trials (lasts in average 2 seconds) although the sound source should be in sight the sound source is marked as not trustworthy. So the robot does not look at it, as long as it persists. Alternatively, a reevaluation of not trusted sound sources is possible after a given time, but experiments revealed that this is not necessary because the speaker verification is working reliable.

To avoid loosing people that are temporarily not in the field of view because of the movement of the robot we developed a short time memory for persons (see figure 6). Our former approach [14] loses persons, who have not generated sensory data for two seconds, no matter what the reason was. Using the terminology of [15] the anchor representation of a person changes from *grounded* if sensor data is assigned, to *ungrounded* otherwise. If the last known person position is no longer in the field of view it is tested whether the person is trusted due to the voice verification described above and whether he was not moving away. If this is applicable the memory will keep the person's position and return to it later according to the attention system. If someone gets out of sight because he is walking away the system will not return to the position. Another exception from the Anchoring described in [16] can be observed if a memorized communication partner reenters the field of view, because the robot shifts its attention to him. It was necessary to add another temporal threshold of 3 seconds since the camera needs approximately 1 second to adjust focus and gain for an acceptable image quality to make a correct face detection possible. If a face is detected within the time span the person remains tracked, otherwise the corresponding person is forgotten and the robot will not look at his direction again. In this case it is assumed that the person has gone while the robot did not pay attention to him.

Besides we added a long time memory that is able to realize, if a person who has left the robot and was no longer tracked, returns during a program run. The long time memory also records person-specific data like name, person height, and size of someone's head to a file. This data is used to increase the robustness of the tracking system. E.g., if the face size of a person is known we can correlate the measured face size with the known one for calculating the person's distance. Otherwise we need to use standard parameters which are not as exact. The application of the long time memory is based on the face identification, which is included in the face detection. To avoid misclassification ten face identification results are accumulated, taking the leading result only if the distance to the second best exceeds a given threshold. After a successful identification it is checked for person specific data stored in a file. Missing values are replaced by the mean of thirty measurements of the corresponding data to consider possible variations in the measurements. Both memory systems are



(a) (b)

Fig. 7. Continuous tracking of a person during object detection: Showing an object to the robot causes the communication partner getting out of sight (a). Nevertheless, the person is successfully tracked (solid) by the person memory that failed (dashed) only once in 11 trials (b).

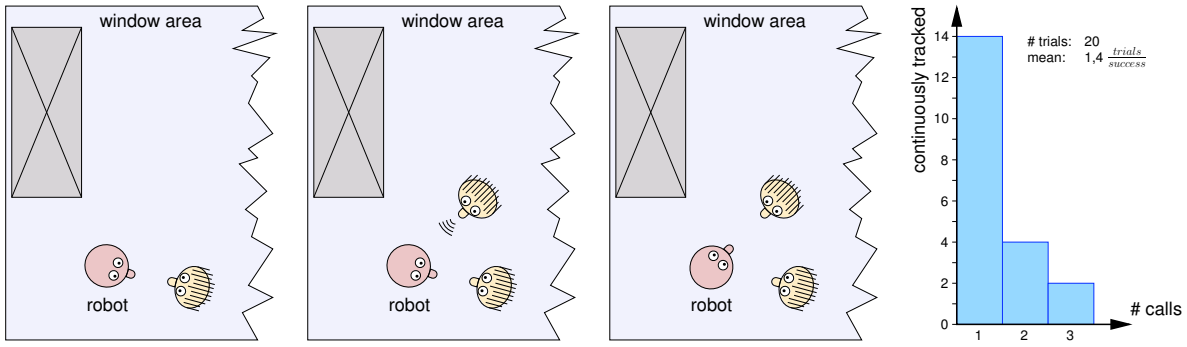


Fig. 8. Person tracking and attention for multiple persons: A person is tracked by the robot (a), a second person stays out of sight and calls the robot (b) to be found and continously tracked (c) without loosing the first person by use of the memory. An auditive signal suffices in most of the trials to achieve this goal (d).

integrated in the tracking and attention module that is running in parallel to the face detection on the the same desktop PC at a frequency of approximately 20Hz. This was estimated in the experiments, which are described in the next section.

VI. Experiments

We developed a system without wide range sensors, only using human-like strategies to observe its surrounding. It is looking for possible communication partners, with a comparable robustness to systems using wide range sensors. To evaluate our approaches we set up different experiments using three main scenarios. For the first scenario one person interacts with the robot testing its short time memory. In the second scenario two people interact with the robot to evaluate the robustness of our tracking for more than one person. Additionally, the attention system is tested. The latter scenario shows to persons interacting with each other, evaluating if the robot control will not disturb a human-human interaction. All experiments were done in an office-like surrounding with common lighting conditions and daylight.

For evaluating the person memory we show objects to the robot as depicted in Fig. 7 (a), resulting in a movement of the robot head that brings the person out of the field of view. After the object detection the robot returns to the last known person position. The tracking was successful in ten of eleven trials as shown in Fig. 7 (b). The failure results from a misclassification of the face detector, since a background very close to the person was recognized as a face and assumed as the primary person. We accept regions close to the memorized position to cope with slight movements of persons.

To evaluate the robustness of the system interaction with several persons we started an experiment with two possible communication partners as described in Figure 8 (a)-(c). In the beginning a person is tracked by XXXX, a second person out of the visual field tries to get the attention of the robot by saying “XXXX, look at me” (Fig. 8 (b)). This should cause the robot to turn to the sound source (Fig. 8 (c)), validate it as voice, and track the corresponding person. The robot was able to recognize and validate the sound as a human voice immediately in 70% of the trials as shown in Figure 8 (d). The probability increased to 90% if a second call was allowed. The robot was always accepting a recently found person for a continuous tracking after a maximum of three calls.

Since the distance of both persons is large enough to prevent them from being in the sight of the robot at the same time we were able to test the person memory. Therefore, we used a basic attention functionality that changes from a possible communication partner to another after a given time interval, if a person did not attempt to interact with the robot. For every attention shift the person memory is used to track and realign with the person temporarily out of view. We observed twenty attention shifts as summarized in Figure 9 reporting two failures and a success rate of 90%.

Subsequently to the successful tracking of humans the ability to react in time to the communication request of a person is required for the acceptance of such a system. Even if the robot can not observe the whole area in front of it. We let two people, who could not be within the field of view at the same time, alternately talk to the robot who

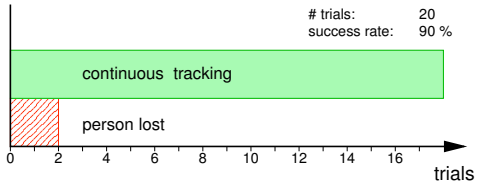


Fig. 9. Use of person memory with several interaction partners: In 90% of the trials a person was successfully tracked and relocated by the person memory (solid). Failed trials are presented dashed.

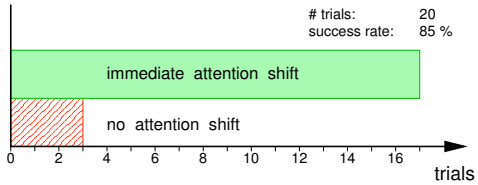
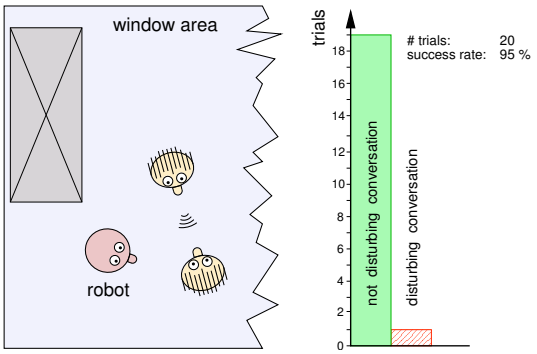


Fig. 10. Attention by speech and analyzing gazing direction: The robot interprets the communication requests from one out of several persons mostly at the first time (solid). In three trials it needed more than one request (dashed).



(a) (b)

Fig. 11. Recognizing the addressee in a talk: Two persons talk to each other in front of XXXX (a) being disturbed by it only once in twenty turn takes (b).

were directly looking at it. In 85% of the trials (see Fig. 10) XXXX immediately recognized who wanted to interact with it using the speaker localization and estimation of the gazing direction by the face detector. Thus, the robot turns towards the corresponding person. Besides the true positive rate of the gazing classification, we wanted to estimate the correct negative rate for verifying the reliability of the gazing classifier, too. Therefore two persons were talking to each other in front of the robot, not looking at it as depicted in Figure 11 (a). Since XXXX should not disturb the conversation of people, the robot must not react to speakers who are not looking at it. The result of this experiment is summarized in Figure 11 (b). Within twenty trials the robot disturbed the communication of the persons in front of it only once. This points out that the gaze classification is both, sufficiently sensitive and selective for determining the addressee in a communication without video cameras observing the whole scene.

The results of the last experiment (see Fig. 12) show that the person memory is not only able to keep a person tracked that is temporarily out of the sensory field, but is fast enough in forgetting people that were gone while the robot was not able to observe them by the video camera. This avoids repeated attention shifts and robot alignments to a 'ghost' person, who is no longer present and accomplishes the required abilities to interact with several persons.

VII. Conclusion

In this paper we presented an approach for detecting and tracking persons with our anthropomorphic robot. We described components for face detection, speaker localization and a tracking module based on Anchoring [15] that avoids the usage of wide range sensors like laser range finders or omnidirectional cameras. Therefore, we developed a method to easily validate sound as voices by taking the results of a face detector in account. To avoid loosing people due to the limited sensory area a short time person memory was developed that extends the existing anchoring of people. Furthermore, a long time memory was added storing person specific data

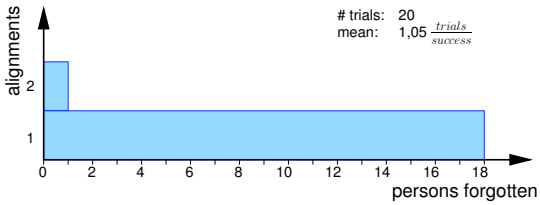


Fig. 12. Number of robot alignments until a disappeared person is forgotten by the person memory: In average 1.05 alignments are necessary.

into file, improving the tracking results. Thus we achieved a robust tracking in real-time with human-like sensors only. It is running on two different robot platforms and is easy portable to other robots using two microphones and a standard video camera.

In the near future we will combine the described modules with a mimic control software for giving the user an intuitive feedback by facial expressions. In addition to mimics we will develop a system able to generate deictic gestures using the robot's arms and hands.

References

[1] A. J. N. van Breemen, "Animation engine for believable interactive user-interface robots," in *Proc. Int. Conf. on Intelligent Robots and Systems*. Sendai, Japan: ACM Press, 2004, pp. 2873–2878.
[2] Anonymized for review, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Edmonton, Canada: IEEE, 2005.
[3] ——, in *Proc. Int. Workshop on Advances in Service Robotics*. Stuttgart, Germany: Fraunhofer IRB Verlag, 2004.
[4] Y. Matsusaka, S. Fujie, and T. Kobayashi, "Modeling of conversational strategy for the robot participating in the group conversation," in *Proc. ISCA-EUROSPEECH2001*, 2001, pp. 2173–2176.
[5] H. Okuno, K. Nakadai, and H. Kitano, "Realizing personality in audio-visually triggered non-verbal behaviors," in *Proc. of the IEEE, Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 392–397.
[6] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, "Generating natural motion in an android by mapping human motion," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1089–1096.
[7] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and S. Behnke, "Integrating vision and speech for conversations with multiple persons," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1295–1300.
[8] Anonymized for review, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, 2005.
[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 511–518.
[10] Z. Zhang, L. Zhu, S. Z. Li, and H. Zhang, "Real-time multi-view face detection," in *Proc. of the IEEE, Int. Conf. on Automatic Face and Gesture Recognition*, Washington, D.C., 2002, pp. 149–154.
[11] D. Giuliani, M. Omologo, and P. Svaizer, "Talker localization and speech recognition using a microphone array and a cross-powerspectrum phase analysis," in *Proc. Internat. Conference on Spoken Language Processing*, 1994, pp. 1243–1246.
[12] S. Coradeschi and A. Saffiotti, "Anchoring symbols to sensor data: preliminary report," in *Proc. of the 17th AAAI Conference*, Menlo Park, California, 2000, pp. 129–135.
[13] Anonymized for review, *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, 2003.
[14] ——, *Pattern Recognition and Image Analysis*, 2004.
[15] S. Coradeschi and A. Saffiotti, "Perceptual anchoring of symbols for action," in *Proc. of the 17th IJCAI Conference*, Seattle, Washington, 2001, pp. 407–412.
[16] Anonymized for review, in *Proc. Int. Conf. on Multimodal Interfaces*. Vancouver, Canada: ACM Press, 2003.

A Methodological Approach relating the Classification of Gesture to Identification of Human Intent in the Context of Human-Robot Interaction*

Chrystopher L. Nehaniv, Kerstin Dautenhahn
Adaptive Systems Research Group
School of Computer Science, University of Hertfordshire
College Lane, Hatfield Herts AL10 9AB, United Kingdom
C.L.Nehaniv@herts.ac.uk

Jens Kubacki, Martin Haegele, Christopher Parlitz
Fraunhofer Institute for Manufacturing
Engineering & Automation (IPA)
Nobelstraße 12
70569 Stuttgart, Germany

Rachid Alami
LAAS/CNRS, Robotics & Artificial Intelligence Group
7, Avenue du Colonel Roche
31077 Toulouse Cedex 4, France

***Abstract*— In order to infer intent from gesture, a broad classification of types of gestures into five main classes is introduced. The classification is intended as a generally applicable basis for incorporating the understanding of gesture into human-robot interaction (HRI). Examples from human-robot interaction show the need to take into account not only the kinematics of gesture, but also the interactional context. Requirements for the operational classification of gesture by a robot interacting with humans are suggested and initial steps in its deployment are discussed.**

***Index Terms*— interaction context, classification of gestures, human-robot activity and interaction.**

I. Introduction: The Need for Classifying Gesture

The word *gesture* is used for many different phenomena involving human movement, especially of the hands and arms. Only some of these are interactive or communicative. The pragmatics of gesture and meaningful interaction are quite complex (cf. [9], [11], [12]), and an international journal [6] now exists entirely devoted to the study of gesture. Applications of service or 'companion' robots that interact with humans, including naive ones, will increasingly require human-robot interaction (HRI) in which the robot can recognize *what* humans are doing and to a limited extent *why* they are doing it, so that the robot may act appropriately, e.g. either by assisting, or staying out of the way. Due to the situated embodied nature of such interactions and the non-human nature of robots, it is not possible to directly carry over methods from human-computer interaction (HCI) or rely entirely on insights from the psychology of human-human interaction. Insights from proxemics and kinesics, which study spatial and temporal aspects of human-human interaction [7], [4], [9] and some insights of HCI, e.g. recognizing the diversity of users and providing feedback acknowledgment with suitable response timing (e.g. [16]), may also prove to be extremely valuable to HRI. Notwithstanding, the nascent field of HRI must develop its own methods particular to the challenges of embodied interaction between humans and robots. New design, validation, evaluation methods and principles particular to HRI must be developed to meet new challenges such as *legibility*, making the robot's actions and behaviour understandable and predictable to a human, and '*robotiquette*', respecting human activities and situations (e.g. not interrupting a conversation between humans or disturbing a human who is concentrating or working intensely — without sufficient cause), as well as respecting social spaces, and maintaining appropriate proximity and levels of attention in interaction. Part of meeting these challenges necessarily involves some understanding of human activity at an appropriate level. This requires the capabilities of recognizing human gesture and movement, and inferring intent. The term "*intent*" is used in this paper in a *limited* way that refers to *particular motivation(s) of a human being that result in a gestural motion directly or indirectly relevant for human-robot interaction*.

In inferring the intent from a human's gesture it is helpful to have a classification of which type of gesture is being observed. Without a sufficiently broad classification, understanding of gesture will be too narrow to characterize what is happening and appropriate responses will not be possible in many cases.

Knowing how to recognize and classify gesture may also serve to inform the design of robot behaviour, including gestures made by the robot to achieve legibility and convey aspects of the robot's state and plans to humans. This in turn will contribute to robot interaction with humans that is legible, natural, safe, and comfortable for the humans interacting with the robot. To begin to approach the complexity of gesture in the context of situated human-

*The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020. This paper extends and supercedes C. L. Nehaniv, "Classifying Gesture and Inferring Intent" *Proc. AISB'05 Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 2005.

robot interaction, the rough classes of gesture described below are developed in order to provide a broad level of description and the first steps toward a pragmatic, operational definition that could be used by an autonomous system such as a robot to help it (1) to infer the intent of human interaction partners, and (2), as an eventual goal, to help the robot use gestures itself (if possible) to increase the legibility of its behaviour.

II. Some Related Work on Recognizing Gesture and Intent

The questions of how gestures are acquired and come to be recognized as meaningful by particular individuals in the course of their development (ontogeny of gesture and its recognition), and conventionalized, elaborated, or lost within particular cultures (evolution of gesture) are large and deep issues, but will not be addressed within the scope in this paper. Psychological/linguistic studies of human gesture use and understanding, related classifications relevant for interaction, language evolution, and language acquisition, e.g. by hearing or deaf children, have all been undertaken (cf. [17]). Such understanding of the development of gesture and its functions may help shed light on gesture in human-robot interaction.

While this paper does not attempt a comprehensive survey of the role and recognition of gesture in human-robot interaction, it does suggest inherent limitations of approaches working with a too narrow notion of gesture, excluding entire classes of human gesture that should eventually be accessible to interactive robots able to function well in a human social environment. Much work with data gloves, typically at a low level for hand gesture recognition for virtual reality or of manipulative grasping has been carried out since the 1990's (e.g. [5], [1]). The important role of gesture for intent communication in human-robot interaction is increasingly being acknowledged, although some approaches still focus only on static hand poses rather than dynamic use of more general types of gesture in context; a survey of hand gesture understanding in robotics appears in [13].

Multimodal and voice analysis can also help to infer intent via prosodic patterns, even when ignoring the content of speech. Robotic recognition of a small number of distinct prosodic patterns used by adults that communicate praise, prohibition, attention, and comfort to preverbal infants has been employed as feedback to the robot's 'affective' state and behavioural expression, allowing for the emergence of interesting social interaction with humans [3]. Hidden Markov Models (HMMs) have been used to classifying limited numbers of gestural patterns (such as grasps or letter shapes) and also to generate trajectories by a humanoid robot matching those demonstrated by a human [2]. Multimodal speech and gesture recognition using HMMs has been implemented for giving commands via pointing, one-, and two-handed gestural commands together with voice for intention extraction into a structured symbolic data stream for use in controlling and programming a vacuuming cleaning robot [8]. Many more examples in robotics exist. Nevertheless, aproaches pursued so far in robotics thus tend to use very limited, constrained, and specific task-related gestural repertoires of primitives, and do not attempt to identify general gestural classes. They have tended to focus on a fixed symbolic set of gestures (possibly an extensible one, in which new gestures can be learned), or focus on only a few representatives from one or two of the gestural classes identified here (e.g. grasps (a subclass of manipulative gestures), or on symbolic and pointing gestures).

Fig. 1. **Gestures with similar kinematics but different functions.** Top row: HELLO (left) an interactional gesture (class 4) is similar to STOP (right) a conventional symbol (class 3). Bottom row: PASS OBJECT (left) is similar and TAKE OBJECT are both multifunctional interactional (class 4) and manipulative (class 1) gestures. Activity and situational context – e.g. stage of interaction and current activity (top row), or location of manipulandum, here a bottle (bottom row) – are be used to disambiguate between such kinematically similar gestures.

III. Insufficiency of Body Model Fitting Analyses: Relating Context to Kinematics.

It should be stressed that a single specific instance of a particular the kind of physical gestural motion could, depending on context and interaction history, reflect very different kinds of human intent. It will not always be possible to infer intent based solely on based the mechanical aspects of human movements (such as changes in joint angles) without taking context into account.

Gestures with identical or near identical human kinematics can be different classes. In general, the kinematic picture alone is not enough to determine the class of a gesture or the human's intent. Examples in shown in figure 1 require contextual information in order to be disambiguated. Relating the context and history of interaction to the kinematics is a key point for recognizing human gestures in HRI. Figure 1 using our classification (see below) illustrates this ambiguity.

IV. Classification of Gestures

To approach this problem, a classification of gesture for inferring intent and assisting in the understanding of human activity should closely relate gesture with limited categories

of intent in situated human activity. The categories of the broad classification presented here thus correspond to and allow the attribution of limited kinds of intent to humans. This classification is developed as an aid for helping robots to achieve limited recognition of situated human gestural motion, so as to be able to respond appropriately if required, while these robots are working in an environment of ambient human activity (such as a home or office), in which, at times, the robots are also assisting or cooperating with the humans. *Applications of this classification will require the mapping of physical aspects of gestural motion in interactional contexts to the five gestural classes (and their subtypes) suggested here.*

The following is a rough, tentative classification. Gestures are classed into five major types with some subtypes.

A. Five Classes (with Subtypes)

1) **'Irrelevant'/Manipulative Gestures.** These include *irrelevant gestures*, *body / manipulator motion*, *side-effects of motor behaviour*, and *actions on objects*. Broadly characterized, manipulation by a human is here understood as *doing something to influence the non-animate environment or the human's relationship to it (such as position)*. Gestural motions in this class are manipulative actions (in this sense) and their side effects on body movement. These 'gestures' are neither communicative nor socially interactive, but instances and effects of human motion. They may be salient, but are not movements that are primarily employed to communicate or engage a partner in interaction. Cases include, e.g. motion of the arms and hands when walking; tapping of the fingers; playing with a paper clip; brushing hair away from the face with the hand; scratching; grasping a cup in order to drink its contents. (Note: it may be very important to distinguish among the subtypes listed above for robot understanding of human behaviour.)
2) **Side Effect of Expressive Behaviour.** In communicating with others, motion of hands, arms and face (changes in their states) occur as part of the overall communicative behaviour, but without any specific interactive, communicative, symbolic, or referential roles (cf. classes 3-5)
   Example: persons talk excitedly raising and moving their hands in correlation with changes in voice prosody, rhythm, or emphasis of speech.
3) **Symbolic Gestures.** Gestural motion in symbol gesture is a *conventionalized signal in a communicative interaction.* It is generally a member of a limited, circumscribed set of gestural motions that have specific, prescribed interpretations. A symbolic gesture is used to trigger certain actions by a targeted perceiver, or to refer to something or substitute as for another signal according to a code or convention. Single symbolic gestures are analogous to discrete actions on an interface, such as clicking a button.
   Examples: waving down a taxi for it to stop; use of a conventional hand signals (a command to halt indicated open flat hand; a military salute); nodding 'yes'; waving a greeting 'hello' or 'goodbye'.
   Note that the *degree of arbitrariness* in such gestures may vary: The form of the gesture may be an arbitrary conventional sign (such as a holding up two fingers with palm forwards to mean 'peace', or the use of semaphores for alphabetic letters). On the other hand, a symbolic gesture may resemble to a lesser or greater extent iconically or, in ritualized form, a referent or activity.
   Further examples: holding up two fingers to indicate 'two'; opening both (empty) hands by turning palms down to indicate a lack of something. Nearly all symbolic gestures are used to convey *content* in communicative interactions.
4) **Interactional Gestures.** These are gesture used to *regulate interaction with a partner*, i.e. used to initiate, maintain, invite, synchronize, organize or terminate a particular interactive, cooperative behaviour: raising a empty hand toward the partner to invite the partner to give an object; raising the hand containing an object toward the partner inviting them to take it; nodding the head indicating that one is listening. The emphasis of this category is neither reference nor communication but on gestures as mediators for cooperative action.[1] Interactional gestures thus concern regulating the form of interactions, including the possible regulation of communicative interactions but do not generally convey any of the content in communication. Interactional gestures are similar to class 1 manipulative gestures in the sense that they influence the environment, but in contrast to class 1, they influence the "animated environment" – *doing something to influence human agents (or other agents) in the environment*, but not by conveying symbolic or referential content.[2]
5) **Referential/Pointing Gestures.** These are *used to refer to or to indicate objects (or loci) of interest* – either physically present objects, persons, directions or locations the environment – by pointing (*deixis* – showing), or indication of locations in space being used as proxies to represent absent referents in discourse. Deictic gesture can involve a hand, finger,

[1]Note that we are using the word "cooperative" in a sense that treats regulating communication or interaction as an instance of cooperation.

[2]Some more subtle examples include putting one's hand on another person's arm to comfort them. Such actions, and others involving physical contact, may be quite complex to interpret as understanding them may require understanding and modeling the intent of one person to influence that state of mind of another. At this point, we class simply them with interactional gestures recognizing that future analysis may reveal deep issues of human-human interaction and levels of complexity beyond the rudimentary types of human intent considered here. A special case worthy of note is human contact with the robot, unless this is directly a manipulation of the robot's state via an interface - e.g. via button presses — which would fall into class 3 (symbolic gesture), non-accidental human contact with the robot is likely to be indicative of an intent to initiate or regulate interaction with the robot (class 4). Physical contact between humans might also involve expression of affection (kissing), or aggression (slapping, hitting) – which generally indicate types of human-human interaction it would be better for a robot to steer clear of!

other directed motion, and/or eye gaze. Checking the eye gaze target of an interaction partner is commonly used to regulate reference and interaction.[3]

Table I summarizes the five classes.

Data on the interaction history and context may help in determining the class of a gesture. If the class is known, then the set of possible gestures can remain large, or be narrowed significantly. Symbolic gestures (class 3) correspond to discrete symbols in a finite set, of which there may be only be a small number according to context or size of the given repertoire of the given symbolic gestural code. Interactional gestures (class 4) are likely to comprise a small, constrained class. Class 1 gestures are either "irrelevant", or to be understood by seeking the intent of the associated motor action or object manipulation (e.g. grasping or throwing an object, arms moving as a side effect of walking). Class 5 (referential and pointing gestures) comprise a very limited class, although pointing can also at times carry affective force (e.g. hostility).

Knowledge of specific conventional codes and signs can help the identification of particular signs within class 3, and also in determining that the gesture in fact belongs to class 3, i.e. is a symbolic communicative signal. Machine learning methods such as Hidden Markov Models may be used successfully to learn and classify gestures for a limited finite set of fixed gestures (e.g. [18]). It seems likely that HMM methods would be most successful with class 3 (symbolic gestures) or within narrow domains within other classes (manipulative grasps with class 1), but how successful they would be at differentiating between classes or for whole classes remains uninvestigated at present.

V. Important Issues

A. Target and Recipient of a Gesture

If a gesture is used interactively or communicatively (classes 2-5), it is important to recognize whether the gesture is directed toward the current interaction partner (if any) — which may the robot, another person (or animal) present in the context, or possibly neither (*target*). If pointing, what is the person pointing to? Who is the pointing designed to be seen by? (*recipient*). If speaking, to whom is the person speaking? If the gesture is targeted at or involves a contact with an object, this suggests it may belong to class 1 (or possibly 5, even without contact). A gesture of bringing an object conspicuously and not overly quickly toward an interaction partner is manipulative (in the sense explained in the discussion of class 1, since an object is being manipulated), but it may well at the same time also be a solicitation for the partner to take the object (class 4). Similarly if the partner has an object, an open hand conspicuously directed toward the partner or object may be a solicitation for the partner to give the object (class 4).

[3]Eye gaze following develops and supports joint attention already in preverbal infants. Language, including deictic vocabulary (e.g. demonstratives such as the words "these" and "that"), and other interactional skills, typically develop on this scaffolding (see [10]).

B. Multipurpose Gestures

It is possible for a single instance of a particular gesture to have aspects of more than one class or to lie intermediate between classes. As mentioned above, handing over an object is both class 1 and 4. And, for example, holding up a yellow card in football has aspects of classes 1 and 3, object manipulation and conventional symbolic signal. Many ritualized symbolic gestures (class 3) also can be used to initiate or regulate interaction (class 4), e.g. the 'come here' gesture: with palm away from the recipient, moving the fingers together part way toward the palm; waving forearm and open hand with palm facing recipient to get attention. More complex combinations are possible, e.g. a gesture of grasping designed by the human to be seen by a recipient interaction partner and directed toward a heavy or awkwardly-sharped target object as a solicitation of the partner to cooperatively carry the object with the gesturer (classes 1, 4, 5).

C. Ritualization: Movement into Classes 3 and 4

Gestures that originate in class 1 as manipulations of the non-animate environment and the person's relationship to it may become *ritualized* to invite interactions of certain types, e.g., cupping the hand next to the ear can indicate that person doing it cannot hear, so that the interaction partner should speak up. Originally cupping the hand near the ear served to improve a person's ability to hear sounds in the environment from a particular direction (class 1), but it may be intended to be seen by a conversational partner who then speaks up (class 4). The hand cupped at the ear can even be used as a conventionalized symbol meaning 'speak up' (classs 3). Other examples of ritualization toward regulation of interaction and also symbolic gesture include mimicking with two hands the motions of writing on a pad as a signal to a waiter to ask for the bill; miming a zipping action across the mouth to indicate that someone should be 'shut up'; or placing a raised index finger over lips which have been pre-formed as if to pronounce */sh/*.

D. Cultural and Individual Differences

Different cultures may differ in their use of the various types of gesture. Some symbolic gestures such as finger signs (e.g. the "OK" gesture with thumb and index finger forming a circle) can have radically different interpretations in other cultures, or no set interpretation depending on the culture of the recipient (e.g. crossing fingers as a sign of wishing for luck, or the Chinese finger signs for some numbers such as 6, 7, 8). Tilting the head back (Greece) or nodding the head (Bulgarian) are used symbolically for 'no', but would certainly not be interpreted that way in many other cultures. Cultures also differ in their types and scope of movement in (class 2) expressive gestures: Consider, for example, the differences of rhythm, prosody, hand motions, eye contact, and facial expressions accompanying speech between British, Italian, Japanese, and French speakers.

Within cultures, differences between different individuals' uses of gestures can be regional, restricted to particular

CLASSIFICATION OF GESTURAL CLASSES AND ASSOCIATED (LIMITED) CATEGORIES OF HUMAN INTENT		
CLASS	NAME	DEFINING CHARACTERISTICS AND ASSOCIATED INTENT
1	'IRRELEVANT' AND MANIPULATIVE GESTURES	INFLUENCE ON NON-ANIMATE ENVIRONMENT OR HUMAN'S RELATIONSHIP TO IT; manipulation of objects, side effects of motor behavior, body motion
2	SIDE EFFECT OF EXPRESSIVE BEHAVIOUR	EXPRESSIVE MARKING, (NO SPECIFIC DIRECT INTERACTIVE, SYMBOLIC, REFERENTIAL ROLE) associated to communication or affective states of human
3	SYMBOLIC GESTURES	CONVENTIONALIZED SIGNAL IN COMMUNICATIVE INTERACTION; communicative of semantic content (language-like)
4	INTERACTIONAL GESTURES	REGULATION OF INTERACTION WITH A PARTNER; INFLUENCE ON HUMAN (OR OTHER ANIMATED) AGENTS IN ENVIRONMENT BUT GENERALLY WITH LACK OF ANY SYMBOLIC/REFERENTIAL CONTENT used to initiate, maintain, regulate, synchronize, organize or or terminate various types of interaction
5	REFERENTIAL/POINTING GESTURES	DEIXIS; INDICATING OBJECTS, AGENTS OR (POSSIBLY PROXY) LOCI OF DISCOURSE TOPICS, TOPICS OF INTEREST; pointing of all kinds with all kinds of effectors (incl. eyes): referential, topicalizing, attention-directing

TABLE I

Five Classes of Gesture. See text for explanation, details and examples. Note that some occurrences of the same physical gesture can be used in different classes depending on context and interactional history; moreover, some gestures are used in a manner that in the same instance belongs to several classes (see text for examples).

social groups within the culture, and vary in particularities (such as speed, repertoire, intensity of movement, etc.) between individuals according to preference or ontogeny. Elderly and young may employ gestures in different ways.

VI. INFERRING THE INTENT OF GESTURE

Being able to identify details of gestural kinematics and even to classify into one of the above classes gives us only starting points for inferring the intent of the person making the gesture due to frequent ambiguity. Resolving this points to the important roles of context and interactional history. Thus, it is necessary to develop operational methods for recognizing the class of gesture in a particular context.[4] If the interactional context of recent activity in which a gesture occurs is known, this can suggest possibilities for which classes (and subtypes) of gesture might be involved. Information on the state of human (e.g. working, thirsty, talking, ...) often can limit the possibilities. Data on the following could help the robot classify the gesture and infer the intent of the human:

(a) the activity of the gesturer is known,
(b) previous and current interaction patterns are remem-

[4]Knowledge of the immediate context in some cases needs to be augmented by taking into account of the broader *temporal horizon* of interactional history (cf. [14]).

bered to predict the likely current and next behaviour of the particular person,

(c) objects, humans and other animated agents in the environment are identified and tracked.

(d) the scenario and situational context are known (e.g. knowing whether a gesture occurs at a tea party or during a card game).

A programme to apply the above classification can be developed as follows: (1) Identify the many, particular gestural motions that fit within each of the five classes. Some gestural motions will appear in more than one class. For example, the same mechanical motion of putting a hand and arm forward with the forearm horizontal and the hand open could indicate preparation to manipulate an object in front of the human (class 1), to show which object is being referred to (class 5), or to greet someone who is approaching, or to ask for an object to be handed over (both class 4). (2) Gestural motions identified as belonging to several classes need to be studied to determine in which contexts they occur: determining in which class(es) particular a instance of the gesture is being used may require consideration of objects and persons in the vicinity, the situational context, and the history of interaction. (3) Systematic characterizations of a physical gestural motion together with interactional contexts in which they are occur could then be used to determine the likely class. (4) Deploy on-board characterizations of the relationships between classes and kinematic gestural motions for a range of typical interactional contexts to infer intent and guide robot behaviour. (5) Updating the Interaction History: Attribution of intent related to gesture can then feedback into understanding of the situational context, including motivational state of the human performing the gesture, and becomes part of the updated interaction history, which can then help in inferring intent from ensuing gestures and activity.

VII. Heuristic-based fast recognition and disambiguation of gestures with a time-of-flight depth sensor

There are mainly three methods to recognize and detect gestures: model-based approaches fit a kinematics model into the scene observed by sensors, recognition based on classifiers use learning algorithms to label gestures, and heuristic-based methods which directly search for hints related to a gesture. Depending on the context of the overall robotic control system, all of these may be of use. The model-based approach is followed by researchers in the Cogniron project, and, as shown above, this must be augmented by contextual and situational knowledge. The goal is to develop algorithms that geometrically fit a model maintained by the robot into the current scene observed by stereo vision systems and a time-of-flight depth sensor proposed in [15]. Apart from this exhaustive approach there is also work related to a computationally much cheaper heuristic-based method only using data delivered by the depth sensor. The motivation for this is two-fold. Firstly, a 'quick' check of the existence of humans in the close vicinity of the robot and a first basic evaluation of possibly important gestures can be used directly for communication. Secondly, outputs of a fast algorithm related for instance to body, head or arm positions can serve to trigger more detailed investigation by e.g. model-based algorithms. Additionally, the data can be used to initialize model fitting.

The heuristic approach first divides the depth scene observed by the sensor into consecutive depth intervals each having a fixed distance and size. The two intervals containing the most measurements are used for binary segmentation of the humans profile. Within the profile the algorithm searches for a human's center point by summing all pixels belonging to the profile and averaging their coordinates. From this point the algorithm searches upwards and determines a bounding box for the head including the neck by incorporating estimates of the shoulder end points. They can be found as being the left and right extremes of the profile at a height roughly at the bottom of the head. Interestingly, the height of the bounding box around the head plus the width of the shoulder can give an estimate of the length of the upper arm as described in medical statistics. Based on this information four cases can be distinguished:

- Outstretched arm away from the body
- Outstretched arm up or down
- Bent arm next to body
- Bent arm in front of body

For each case further heuristic algorithms are used to determine the hand position and orientation. This can be used to recognize and discriminate between basic gestures. See Figure 2 for a visualization of how the program finds a WAVE gesture and SHAKE HANDS gesture; both are interactional gestures (class 4). By using additional information such as orientation and distance of the human towards the robot and internal state of the robot, tentative disambiguations between similar gestures have been made.

VIII. Conclusions, Next Steps and the Future

In order to infer the intent of a human interaction partner, it may be useful to employ a classification of gesture according to some major types – five in the tentative classification proposed here – whose intent may be (1) absent / directed to objects or environment, (2) incidentally expressive, (3) symbolic, (4) interactional, or (5) deictic. A summary of the classes is given by Table I.

In order to deploy the inference of intent on robots interacting with humans it is necessary to operationalize the distinctions between these (sometimes overlapping) classes. This may require the use of knowledge of human activity, recognition of objects and persons in the environment, and previous interactions with particular humans, as well as knowledge of conventional human gestural referencing and expression, in addition to specialized signaling codes or symbolic systems.

Work in Cognrion now focuses on the organization of the robot decisional abilities and more particularly on the management of human interaction. There is explicit

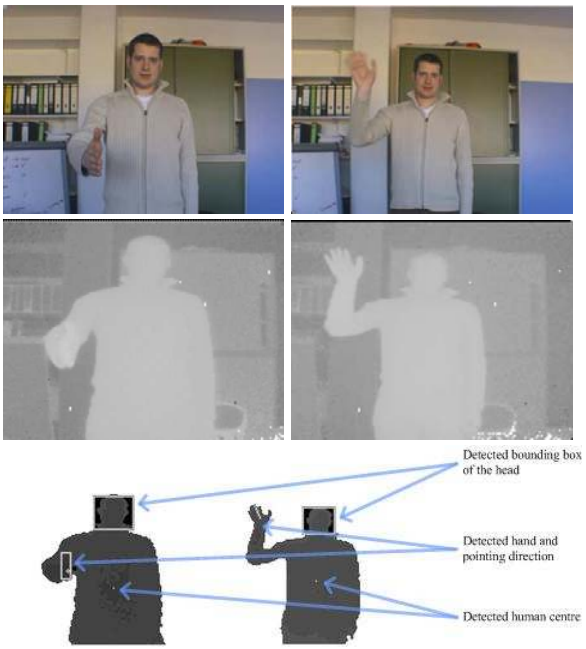


Fig. 2. **Heuristic-based fast recognition and disambiguation of interactional gestures.** Detecting bounding box of head, hand and its orientation, and center of human (top row: color image of gesturing human in robot's vicinity. second row: depth image derived from camera images. Left column: SHAKE HANDS gesture, right column: WAVE gesture.

management of the interactions between the robot companion and its human partners. This requires essentially task-oriented processes that each consist of establishing a common goal, achieving it and verifying commitment of all agents involved during the task performance. Indeed, perception of the human partner is one essential source of information all along the human-robot interaction process from the detection of human presence to the monitoring of human activity and the continuous estimation of its commitment level to a joint goal. This viewpoint is compatible with and served by the classification of gestures proposed here. It also helps us to operationalize use of the classification. Indeed, gestures of type 3 and many of type 1 may be considered as task-oriented and the inference of their intent can be done relative to the task at hand. Gestures of type 4 include generic interactional gestures that may serve to manage the session itself: inviting the robot to start an interaction, suspending or stopping an interaction session, etc. Many gestures of type 4 are consequently task independent.

The classification presented here suggests some requirements for the design and implementation of systems inferring intent from gesture based on this classification. These requirements might be realized in a variety of different ways using, e.g. continuous low-key tracking or more detailed analysis, event-based and/or scenario-based recognition, and prediction of human activity based on models of human activity flows (with or without recognition of particular humans and their previous interactions), depending the particular needs of the given human-robot interaction design and the constraints and specificity of its intended operational context. Design of a robot restricted to helping always the same user in the kitchen environment would be quite different from one that should be a more general purpose servant or companion in a home environment containing several adults, children and pets, but the classification presented here is applicable in informing the design of gesture recognition for inferring intent in either type of system, and for designing other HRI systems.

Finally, effective human-robot interaction will require generation of gestures and feedback signals by the robot. The classification given here can suggest categories of robotic gestures that could be implemented to improve the *legibility* to humans of the robot's behaviour, so that they will be better able to understand and predict the robot's activity when interacting with it.

REFERENCES

[1] BERNADIN, K., OGAWARA, K., IKEUCHI, K., AND DILLMANN, R. A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Trans. on Robotics 21*, 1 (2005), 47–57.

[2] BILLARD, A., EPARS, Y., CALINON, S., CHENG, G., AND SCHAAL, S. Discovering optimal imitation strategies. *Robotics & Autonomous Systems,* Special Issue: Robot Learning from Demonstration *47*, 2-3 (2004), 69–77.

[3] BREAZEAL, C., AND ARYANANDA, L. Recognition of affective communicative intent in robot-directed speech. *Autonomous Robots 12*, 1 (2002), 83–104.

[4] CONDON, W. S., AND OGSTON, W. D. A segmentation of behavior. *Journal of Psychiatric Research 5* (1967), 221–235.

[5] DUBOIS, E., NEDEL, L. P., FREITAS, C. M. D. S., AND JACON, L. Beyond user experimentation: notational-basd systematic evaluation of interaction technieuqs in virtual reality environments. *Virtual Reality 8*, 2 (2005), 118–128.

[6] *Gesture*. ISSN: 1568-1475. John Benjamins Publishing Co., The Netherlands `http://www.benjamins.com/cgi-bin/t_seriesview.cgi?series=GEST`.

[7] HALL, E. T. *The Dance of Life: The Other Dimension of Time*. Anchor Books, 1983.

[8] IBA, S., PAREDIS, C. J. J., AND KHOSLA, P. K. Interactive multimodal robot programming. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington D.C., May 11-15, 2002* (2002).

[9] KENDON, A. Movement coordination in social interaction: Some examples described. *Acta Psychologica 32* (1970), 100–125.

[10] KITA, S., Ed. *Pointing: Where Language, Culture and Cognition Meet*. Lawrence Erlbaum Associates, Inc, 2003.

[11] MEY, J. *Pragmatics: An Introduction*. Blackwell Publishers, 2001.

[12] MILLIKAN, R. G. *The Varieties of Meaning: The 2002 Jean Nicod Lectures*. MIT Press/Bradford Books, 2004.

[13] MINERS, W. B. *Hand Gesture for Interactive Service Robots*. MSc Thesis, The University of Guelph, Faculty of Graduate Studies, August 2002.

[14] NEHANIV, C. L., POLANI, D., DAUTENHAHN, K., TE BOEKHORST, R., AND CAÑAMERO, L. Meaningful information, sensor evolution, and the temporal horizon of embodied organisms. In *Artificial Life VIII* (2002), MIT Press, pp. 345–349.

[15] OGGIER, T., AND LANG, G. K. User manual: Swissrangertm 2, December 2003, Rev. March 2004.

[16] SHNEIDERMAN, B. *Designing the User Interface: Strategies of Effective Human-Computer Interaction*, 3rd ed. Addison-Wesley, 1998.

[17] VOLTERRA, V., CASELLI, M. C., CAPIRCI, O., AND PIZZUTO, E. Gesture and the emergence and development of language. In *Beyond Nature-Nurture: Essays in Honor of Elizabeth Bates*, M. Tomasello and D. I. Slobin, Eds. Lawrence Erlbaum Associates, 2004.

[18] WESTEYN, T., BRASHEAR, H., ATRASH, A., AND STARNER, T. Georgia tech gesture toolkit: Supporting experiments in gesture recognition. In *ICMI 2003: Fifth International Conference on Multimodal Interfaces* (2003), ACM Press.

Classifying Human Activities in Household Environments

Stefan Vacek, Steffen Knoop, Rüdiger Dillmann
Institut for Computer Design and Fault Tolerance
University of Karlsruhe
D-76128 Karlsruhe, Germany
{vacek, knoop, dillmann@ira.uka.de}

Abstract

The recognition of daily human actvitities is becoming more and more important for humanoid robots. For the robot, being a daily companion of the human, it is crucial that it is able to understand what the human is doing in order to react accordingly.

Although there exist many systems for recognising human activities there is a lack of having a structured classification of these activities. In this paper different concepts for classifying human activities are presented. First, a concept motivated by recognition approaches is presented, which is called structural classification. The second concept is guided by the functional meaning of activities. These two concepts are then combined in a third classification which connects the structural with the functional classification. The benefit of the combined classification is, that it adds semantics into the structural view of activities and it enables the algorithms for further refinement of the recognition.

1 Introduction

One investigated field of application of mobile robots is within the environment of humans. For the robot, being a companion of the human in household environments, it must have several abilities. These include supporting or helping the human, interacting with the human, learning skills and tasks or recognising the human's intention. While the detection and tracking of people is the first step for the robot of being aware of humans in its surrounding, it is important to understand what the human is doing.

The aim of this paper is to investigate concepts for designing a classification of human activities in household environments. This classification supports several research activities and has multiple benefits, which are:

- It serves as a basis for the recognition of activities and can be used several algorithms. Furthermore with this classification it is possible to introduce semantic knowledge into the recognition.
- It establishes a system wide common taxonomy about human activities which can be used widely. Especially for a humanoid robot, this knowledge can be used in:
  - Dialogues, by helping to understand the users actions and recognising his or her willingness to interact.
  - Learning skills and tasks from a human while observing the demonstrator.
  - Situation awareness and intentionality, by understanding the human's activity which is part of the actual situation and recognising his or her intention.
- It helps to build a semantic link between the robot's own abilities and the activities of humans. Thereby it supports the robot to reason about its own abilities and to decide whether and how it can help the human.

It is obvious, that not all possible human activities can be classified. The reasons are that the kind of classification always depends on its purpose and each field of application has its own interests which cannot be covered completely in an overall classification. Therefore, the presented classifications concentrate on a subset of possible human activities: activities in household environments. Its purpose is towards the use in a humanoid mobile robot being a daily companion of the human.

In the following section a brief overview of the state of the art is presented, in section 3 a general introduction on human activities is given and a categorisation of classifying activities is described. The succeeding subsections explain the different concepts of classifying human activities. The combined structure, depicted in section 3.3, incorporates the presented approaches into one classification.

2 Related work

Most of the researchers do not define an explicit classification of human activities. In fact most publications concentrate on detection, recognition and interpretation.

Sierhuis et al. [Sierhuis *et al.*, 2000] describe a representation of work practice which consists of activities of the involved people. Work is defined as transforming input to output. An activity is more than that, namely it includes also collaboration between individuals. An activity is described by how, when, where and why an activity is performed and identify the affects of an activity. Activities locate behaviour of people and their tools in time and space.

In [Rao and Shah, 2001] a flat list of captured actions is used. The recognition evaluates the position of the hand in order to interpret the resulting trajectories. [Sukthankar and Sycara, 2005] uses an acyclic graph to model a specific behaviour. Each edge consists of a basic body motion together with an environmental feature.

Lokman and Kaneko [Lokman and Kaneko, 2004] presented a hierachical structure of the body-parts and joints to derive a classification of human actions. The basic ideas are, that the human does not always use all body-parts for an activity and that multiple actions could happen simultaneously.

A hierarchical structure of actions is used in [Mori *et al.*, 2004] where the actions are classified in a tree-like structure. An action is modelled by Continous Hidden Markov Models. The recognition starts at the root node and for all child nodes, the likelihood is calculated. If there is a valid child, the recognition descends in this lower level and the recognition starts again. If no valid child can be found, the recognition stops. At each level of the tree, there is a special node, called "etc" which denotes "every other" action, not listed in the tree at that level. For example at the first level, there are "Sitting", "Lying", "Standing" and "Etc".

In [Kojima *et al.*, 2002] a concept hierarchy of body actions is used for extracting a natural language description of human actions out of image sequences. An activity is represented by a so called "case frame" where a case frame expresses the relationship between cases in a natural sentence (like *agent, object, locus, source*, etc.). The hierarchy of actions starts at a generic level and is refined at each level by introducing additional values into the case frame. These additional values correspond to extracted image features. E. g. *be* becomes *move* by introducing the speed of the torso and therefore replacing the verb.

A similar approach is used in [Herzog and Rohr, 1995]. Here, an activity is represented in terms of predicate logic. Each term then describes an action with specific attributes which can be further refined (e.g. "move" + "fast" becomes "running").

Patterson et al. [Patterson *et al.*, 2003a] use RFID-tags to observe the user's interaction with objects. The activity models should be human understandable and that they describe the activities intuitively. An activity is described by a set of touched objects. For recognising an activity they use Dynamic Bayesian Networks.

Different aspects of modelling and recognising human behaviours are present in [Liao *et al.*, 2004]. Modelling human behaviour comprises the decomposition of behaviours and the abstraction and thus the grouping of behaviours. A big problem in human behaviour recognition is the gap between the raw sensor data and the recognition algorithms. In [Patterson *et al.*, 2003b] GPS data is used to infer about the user's movement within a city and his transportation mode (i.e. by bus, by car or by foot). A particle filter is used to estimate the state of the user.

In [Bui, 2003] a framework for probabilistic plan recognition of hierarchies of activities is presented. So called Abstract Hidden Markov Memory Models are introduced which allow to estimate sub-policies depending on the previous history of the process. The system is demonstrated in an office monitoring scenario where different actions like "going to the printer" are recognised.

Another approach for hierarchical modelling is presented in [Pynadath and Wellman, 2000]. A PSDG (probabilistic state-dependent grammar) is used to define plans and subplans. Parsing a given observation results in probabilities for different subplans allowing the recognition of actions.

3 Classification of human activities

Before defining a classification of human activities, it has to be made clear what the term "activity" stands for. Following dictionaries (e.g. [dictionary.com, 2005]), they state:

Definition 1 activity*: "state of being active"*

Looking into the more specific term *human activity*, dictionaries (s. e.g. [WordNet 2.0, 2005]) define it as:

Definition 2 human activity*: "something that people do or cause to happen"*

It is clear that it is not possible to classify *all* existing human activities. In fact a classification for only a subset, namely activities in household environments, is presented. Beside looking into typical household scenarios, the demands arising from the *COGNIRON* project (the *cognitive robot companion*, s. `http://www.cogniron.com`) were taken into account.

Typical activities are:

- Talking to someone
- Walking around
- Sitting on a chair
- Taking out a beer from the fridge
- Opening a door
- Grasping a cup
- Placing a cup on a saucer

This list isn't complete, it should only give an impression about the variety of human activities in typical household scenarios. Indeed, these activities can also be combined like *walking while talking to someone*.

For designing the classification, some important issues have to be considered:

- The classification should not depend on any existing algorithm doing activity recognition but it must also be possible to use this classification for the development of future recognition algorithms.
- It should be open ended in a way that new categories could be added in the future and also previously unconsidered activities should be categorised later on.
- It should have a clear structure for the ease of usage.
- It should be usable for different disciplines, like computer vision, dialogues or task learning.

Therefore different concepts of classifications were investigated following different approaches. The first one is derived from the *structure* of the human body, that is, each activity

is classified based on the body parts which are *used* for this activity (”How is the activity performed”). The second one is guided by the *functional* meaning of the activities. That is, the semantics of an activity is classified according to its function (”What is the aim of the activity”). Finally these two concepts are incorporated into one where the two previous concepts (structural vs. functional) are orthogonal.

The following subsections describe these concepts in detail.

3.1 Classification by structure

As has been mentioned before, classifying activities by structure means that each activity is categorised based on the involved *structures*. The term *structure* is meant to be a body part, the whole person, an object or a place. The classification is an algorithmic guided approach, because many algorithms evaluate the pose and motion of certain body-parts in order to recognise the activity (e.g. [Aggarwal and Cai, 1999]. It starts with groups of activities belonging to single body-parts and creates new groups by combining groups.

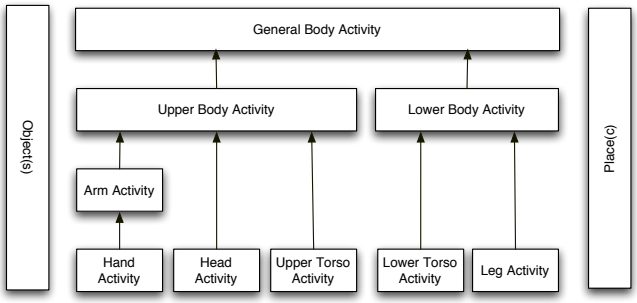


Figure 1: Overview of human activities classified by structure

Figure 1 shows the identified groups of the classification. Each part describes, which structure is involved in this particular group. The arrows, going from one part to another (e.g. from ”Head activity” to ”Upper Body Activity”), denote dependencies of body parts required for this (higher level) group of activities. In the case of ”upper body activity” not all incoming parts need to be active in order to form a valid activity.

The two groups *object* and *place* play a special role in a way that they can augment the meaning of each part. For example, ”hand activity” together with ”object” form a new group of activities. Another example is the command ”put that there” where an object and a place are involved.

It is clear, that a group like ”arm activity” contains a lot of activities like all arm gestures, grasps, etc. Therefore, this classification is a very coarse one, but it helps in understanding how a specific activity is performed or to be more precise, which parts are involved in an activity.

3.2 Classification by function

In contrast to the previously described structural classification, the *classification by function* is guided by the purpose or aim of an activity. In cognitive psychology, human activity is characterized by three features [Anderson, 1989]:

Direction: Human activity is purposeful and directed to a specific goal situation.

Decomposition: The goal that is to be reached is being decomposed into subgoals.

Operator selection: There are known operators that may be applied in order to reach a subgoal. The concept *operator* designates an action that directly realizes such a subgoal. The solution of the overall problem is representable as a sequence of such operators.

Humans tend to perceive activity as a clearly separated sequence of elementary actions. Therefore the set of supported elementary actions is derived from human activity mechanisms. Based on the purpose that is being aimed at by the activity, a classification into two categories is appropriate:

Performative activities: These activities aim at reaching a certain goal in terms of fulfilling a task, they change the state of the human or the state of his or her environment like walking around or grasping an object.

Interaction activities: This class does not only comprise activities within a dialogue, but also for enhancing the learning of demonstrated tasks and guiding the robot.

Figure 2 shows the overall classification based on the modality of their application. Performative and interactive activities are explained in more detail in the following subsections.

Performative Activities

Manipulation, navigation and the utterance of verbal performative sentences are classified as performative activities.

Manipulation: During object manipulation, grasps and movements are relevant for interpretation.

- **Grasps:** For the classification of grasps that involve one hand established schemes can be reverted to. Here, an underlying distinction is made between grasps that do not need to change finger configurations while holding an object until placing it somewhere (”static grasps”) and grasps that require such configuration changes (”dynamic grasps”). While for static grasps exist exhaustive taxonomies based on finger configurations and the geometrical structure of the carried object, dynamic grasps may be categorized by movements of manipulated objects around the local hand coordinate system. Grasps being performed by two hands have to take into account synchronicity and parallelism in addition to single grasp recognition.
- **Movement:** Here, the transport of extremities and of objects has to be discerned. The first may be further partitioned into movements that require a specific goal pose and into movements where position changes underly certain conditions (e.g. force/torque, visibility or collision). On the other hand, the transfer of objects can be carried out with or without contact. It is very useful to check if the object in the hand has or has not tool quality. The

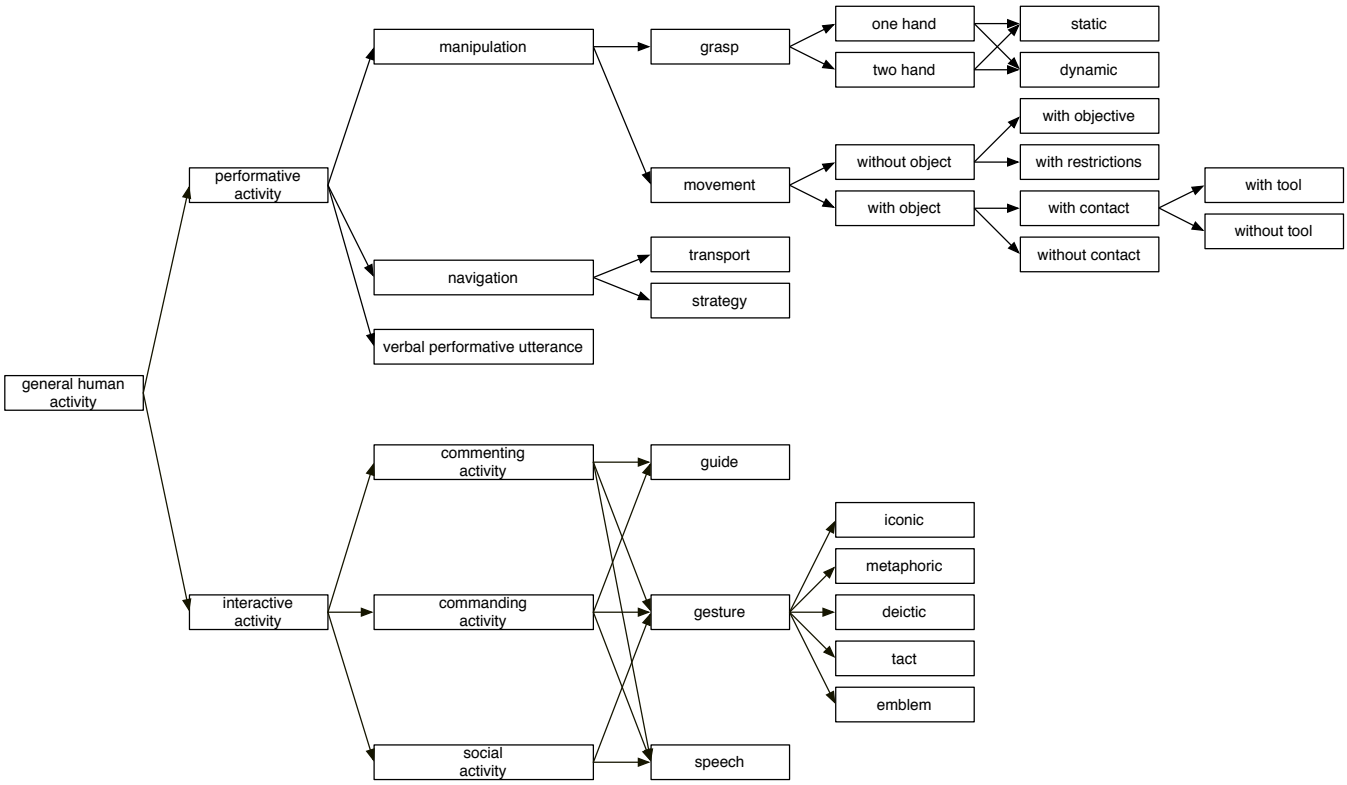


Figure 2: Overview of human activities classified by function

latter case eases reasoning on the goal of the operator (e.g.: *tool type* screwdriver → *operator* turn screw upwards or downwards).

Navigation: In contrast to object manipulation, navigation means the movement of the human himself. This includes position changes with a certain destination in order to transport objects and movement strategies that may serve for exploration.

Verbal performative utterance: In language theory, utterances are performative if the speaker is performing the activity he is currently describing. This could help robotic systems to understand the actual activity.

As can be seen by the complexity of grasp performance or navigation, observation of performative actions requires vast and dedicated sensors. Hereby, diverse information is vital for the analysis of an applied operator: a grasp type may have various rotation axes, a certain flow of force/torque exerted on the held object, special grasp points where the object is touched etc.

Interaction Activities

Commenting, commanding and social interaction are classified as interaction activities. They are not only performed using speech but also gestures with head and hands belong to these categories.

Commenting activities: Humans refer to objects, places and processes by their name, they label and qualify them. Primarily, this type of action serves for enhancing dialogues and it also helps for learning and interpreting.

Commanding activities: Giving orders falls into the second category. This could be e.g. commands to move, stop, hand over or even complex sequences of single commands, that directly address robot or human activity.

Social activities: This class is mainly intended at exchanging information. It includes activities like greeting or asking.

It is clear, that in contrast to the structural classification, a single activity of a body-part can result in activities of different groups. For example an activity with the hand can be a grasping activity or a commanding gesture. Furthermore, a single activity can be achieved with different body-parts, for example affirmation (a commenting activity) can be done with the hand (”thumbs up”) as well as with the head (”nodding”).

3.3 Combining the classifications

The problem of the two presented classifications is, that they consider mainly one dimension of concepts for classifying human activities. To be more precise, the structural classification is based on *how* (or *which body-part*) an activity can be detected and therefore classified. On the other hand, the functional classification mainly concentrates on *what* type of activity is present without considering which body-parts are involved (and therefore need to be algorithmically evaluated).

So the question is how to create a classification which connects the *how* and the *what* type. Or, in other words, how to fill the gap between semantic and detection.

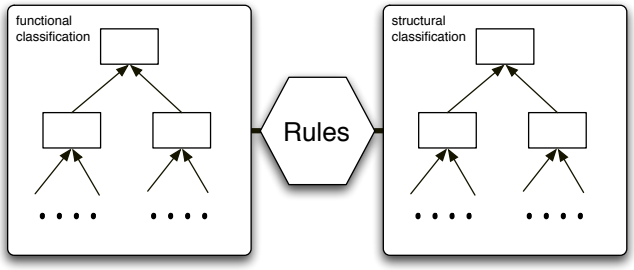


Figure 3: Connecting the structural with the functional classification.

The idea is to introduce a set of rules (s. figure 3) which connects certain structural parts with the corresponding functional group. The connection is bi-directional giving information of the structural into the functional classification and vice versa.

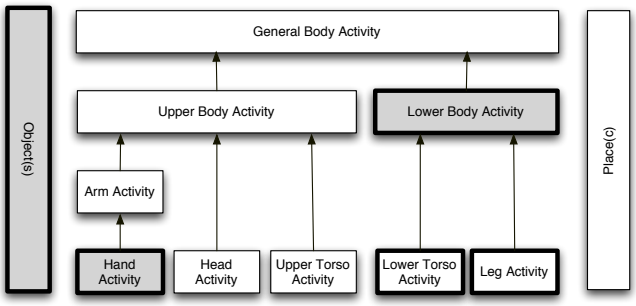


Figure 4: The relevant structural groups for the example "transport of an object".

We illustrate the idea using the activity "transporting an object" as an example. In the structural classification, the groups "Object(s)", "Hand Activity" and "Lower Body Activity" are involved, which can be detected with appropriate sensors. This is depicted in figure 4. The set of rules, which holds the background knowledge about mapping between structural and functional representations, activates the corresponding activities in the functional model.

At this point, the hierarchical form of the functional classification enables further reasoning about the performed activities and their more generalised activity classes. In the given example, it is now possible to derive the classes "one hand", "grasp", "manipulation", etc. The advantage of having the more generalised classes is, that others could use the information at the level of detail they need. For example, if the dialogue only wants to know, if there is a performative activity, the requested information can easily be delivered.

Additionally, the knowledge of the functional classification allows also for refining the detected activity. More features can be extracted by the perception in order to evaluate if there is a more specific activity. Also, the current context can be used for further refinement.

4 Conclusion and Future Work

In this paper we presented different concepts for classifying human activities. The idea is to establish a common taxonomy for recognising activities as well as using it in other applications. The classification was done for activities in household environments to help humanoid robots in recognising human activities. The first classification is based on the body structure of the human being, which is also motivated by algorithmic approaches. The second classification is structured based on the functional meaning of a human activity, bringing semantics into the classification. These two classification are then combined into a third classification which connects the structural (body-part driven) view with the functional view.

The next steps are to further validate the proposed classification and to continue with the classification in terms of extending it with new activities. For validating the classification an activity recognition will be developed which will be used to teach the robot and to detect the users intention in order to enable the robot to assist the human. Additionally, social studies about human behaviour in the presence of robots will be investigated. The set of rules will be developed in order to establish the connection between the structural and the functional classification. Furthermore, investigations will be done, how the rules can be learned in order to reduce the required a priori knowledge.

Acknowledgments

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion", s. `http://www.cogniron.com`) and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

[Aggarwal and Cai, 1999] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.

[Anderson, 1989] J. Anderson. *Kognitive Psychologie, 2. Auflage*. Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, 1989.

[Bui, 2003] Hung H. Bui. A general model for online probabilistic plan recognition. In *International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 9-15 2003.

[dictionary.com, 2005] dictionary.com. Definition of activity, 2005.

[Herzog and Rohr, 1995] Gerd Herzog and Karl Rohr. Integrating vision and language: Towards automatic description of human movements. In *Proc. of the 19th Annual German Conference on Artificial Intelligence (KI-95)*, Bielefeld, Germany, Sept. 11-13 1995.

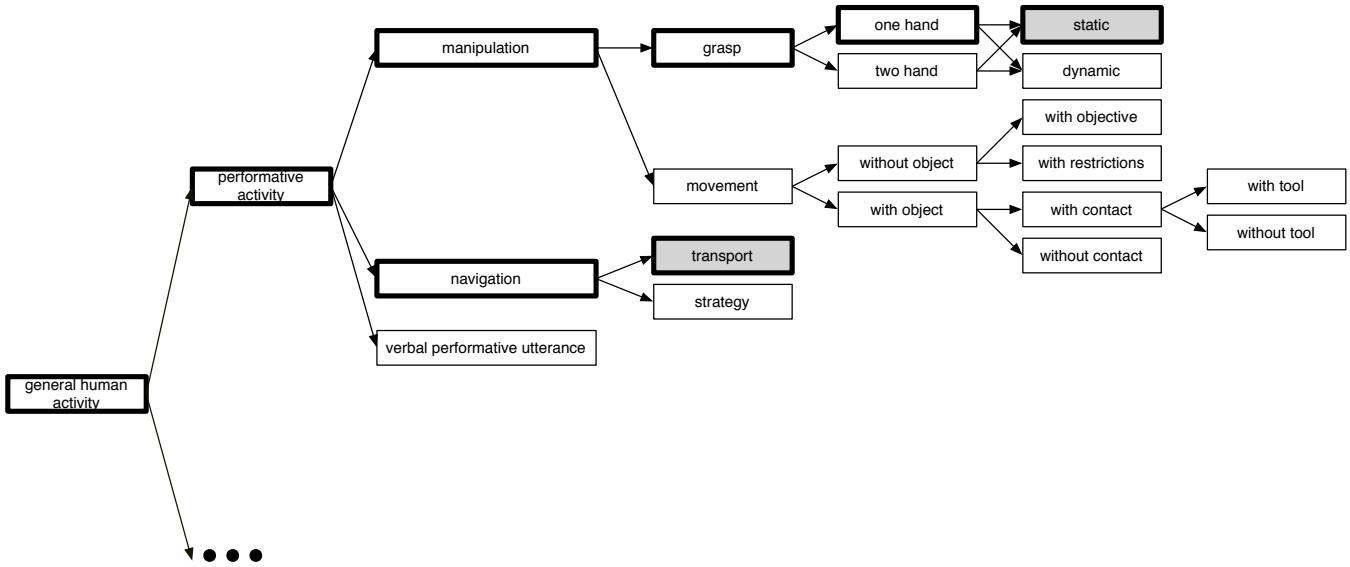


Figure 5: The relevant structural groups for the example "transport of an object".

[Kojima *et al.*, 2002] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activites from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 2(50):171–184, 2002.

[Liao *et al.*, 2004] Lin Liao, Don Patterson, Dieter Fox, and Henry Kautz. Behavior recognition in assisted cognition. In *The AAAI-04 Workshop on The AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, California, USA, July 25 2004.

[Lokman and Kaneko, 2004] Juanda Lokman and Masahide Kaneko. Hierarchical interpretation of composite human motion using constraints on angular pose of each body part. In *13th IEEE Int'l Workshop on Robot and Human Interactive Communication (RO-MAN 2004)*, pages 335–340, Kurashiki, Okayama Japan, Sept. 20-22 2004.

[Mori *et al.*, 2004] Taketoshi Mori, Yushi Segawa, Masamichi Shimosaka, and Tomomasa Sato. Hierarchical recognition of daily human actions based on continous hidden markov models. In *Proc. of the Sixth IEEE Int'l Conf. on Automatic Face and Gesture Recognition (FGR'04)*, Seoul, Korea, May 17-19 2004.

[Patterson *et al.*, 2003a] Don Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Expressive, tractable and scalable techniques for modeling activities of daily living. In *Proceedings of UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, Seattle, Washington, USA, October 12, 2003 2003.

[Patterson *et al.*, 2003b] Donald J. Patterson, Lin Liao, Dieter Fox, and Henry Kautz. Inferring high-level behavior from low-level sensors. In *Proc. of the International Conference on Ubiquitous Computing (UbiComp)*, pages 73–89, Seattle, Washington, USA, October 12-15 2003.

[Pynadath and Wellman, 2000] David V. Pynadath and Michael P. Wellman. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence*, pages 507–514, Stanford, California, USA, June 30 - July 3 2000.

[Rao and Shah, 2001] Cen Rao and Mubarak Shah. View-invariance in action recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume II, pages (II)316 – (II)322, Kauai Marriott, Hawaii, Dec. 9-14 2001.

[Sierhuis *et al.*, 2000] Maarten Sierhuis, William J.Clancey, Ron van Hoof, and Robert de Hoog. *Modelling and Simulating Human Activity*. AAAI Fall Symposium on Simulating Human Agents., 2000.

[Sukthankar and Sycara, 2005] Gita Sukthankar and Katia Sycara. A cost minimization approach to human behavior recognition. In *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, to appear 2005.

[WordNet 2.0, 2005] WordNet 2.0. Definition of human activity (http://www.cogsci.princeton.edu/cgi-bin/webwn2.0?stage=1&word=human+activity), last visited: 2005/01/11, 2005.

Extraction, Evaluation, Selection and Classification of Motion Features for Human Activity Recognition

Steffen Knoop, Stefan Vacek and Rüdiger Dillmann
Industrial Applications of Informatics and Microsystems
Institute of Computer Science and Engineering
University of Karlsruhe, Germany
Email: {knoop,vacek,dillmann}@ira.uka.de

Sebastian Brännström and Henrik I. Christensen
Centre for Autonomous Systems
Royal Institute of Technology (KTH)
SE - 100 44 Stockholm, Sweden
Email: teknolog@gmail.com, hic@nada.kth.se

***Abstract*— This report describes the work performed at University of Karlsruhe concerning *Human Activity Recognition* in the context of the EU project *Cogniron*.**

Within another part of the project, human motion and body configuration is observed and tracked. This *intrinsic* information is used together with *extrinsic* features to recognize and classify human activities. Recognition is done using a 3-step approach: (1) From human body motion, a feature set is extracted. (2) These features are evaluated regarding their *relevance* for each activity, and (3) for each activity, a FFNN classifier computes a likelihood based on manually segmented training data sets.

This report gives a state of the art, and then describes in detail the depicted activity recognition approach. Experiments and results are given which show that this method works with over 100 features and more than 10 activities.

I. Introduction

Humans communicate to a large extent through physical movement of their limbs. Many activities can be readily recognized just by observing the motion of the limbs of the human body, or even the motion of the entire body.

How one person interacts with another is in many cases highly dependent on the observation of what the other person is doing. For example, we do not try to shake hands with a person running quickly past us.

When designing a socially interacting system such as a service robot, it is desirable to try to mimic this ability. Not only does activity recognition by the robot greatly improve its ability to interact, but it also tends to increase the level of comfort for people with which it interacts.

Apart from transmitting important activity information, limb movements are an important accessory when communicating. The motions known as *gestures* generally make up a large part of the information flow when two people are talking to each other. Similarly for a robot, accurate activity recognition could provide a helpful context for its speech recognition system.

It is therefore clearly desirable for a socially interacting robot to be able to interpret observed motions, and recognize the activities that cause them.

II. Background

Activity recognition is a highly active research field and the amount of published material is immense. Good starting points to the area are the comprehensive surveys composed by Cédras and Shah [1] and by Gavrila [2], and the slightly shorter review by Aggarwal and Cai [3]. Additionally, Wang, Hu and Tan [4] cover some work done after 2000.

A large field of application for activity recognition is given by the problem of video surveillance, e.g. in public areas where often surveillance cameras already exist. The topic of surveillance rises already several restrictions, which include usage of cameras only, or large distance between sensor and target. In most surveillance cases, the background can even be assumed to be static, which is an important factor for recognition and can also affect the method and algorithm selection. Activity recognition systems for surveillance applications have been developed e.g. by Ribeiro et al. [5] [6] and Nascimento et al. [7], which rely on large area camera images. These approaches use the trajectory of the whole person for activity classification, and do not use the body configuration. This is on the one hand due to the fact that the observed activties are large-scale and can thus be classified by observing the whole body trajectory over time, on the other hand, the sensor data simply can not provide such detailed information as it would be needed for determination of the body configuration. Also, in a surveillance context, the system is only directed at observation, without any active components.

These vision-based activity recognition approaches still follow a common methodology: From the raw input data (camera images in this case), a set of features is extracted which is in a second step processed to classify performed human activities. Although the aim and conditions are different for the context of this work, it still follows a similar approach.

One general trend has been to transfer successfull techniques from the typically simpler problem of speech recognition to the more difficult problem of activity recognition. The impressive results within that field and its rapid convergence to a handful of techniques suggests similar approaches to activity recognition.

Activity recognition is preferably divided into several independent or weakly dependent sub-tasks. As is common in rapidly advancing research fields, a lot of different terms are used by different authors to describe essentially the same things.

In this report we shall divide the process into three tasks. The first is *motion capture*, which covers the entire problem

of observing a human subject and obtaining a digital representation. The second is *motion analysis*, where the motion capture data is processed to make it suitable for the third step, the actual *recognition* or *classification* itself.

In general terms, we may then describe the recognition system as consisting of the tree steps below:

1) Observation of the human subject and digitalization of the motions *(motion capture)*. This can either be done with view-based approaches, retrieving abstract feature patterns, or based on kinematic models of the human, which results in 2d or 3d motion trajectories of the human body DoFs.
2) Analysis of the motion and pattern construction *(motion analysis)*. There are two main approaches: Segmentation into basic elements, often called *motion primitives* by recognition of key points, or concurrent feature extraction for a later analysis.
3) Comparison with previously stored patterns *(recognition)*. This task is commonly solved using a classifier, which in most cases has been trained with a manually segmented data set. This classifier can either include an explicit time model (like HMMs, Bayesian nets) or be time-independant (like Neural Networks, Support Vector Machines etc.).

III. Human activities

Talking about human activity recognition first raises the question for a definition of human activities. To our knowledge, there has been no closed-form definition for human activities so far in literature within the area of activity recognition, and many authors use examples to describe the desired granularity and features of activities.

Ribeiro et al. [5] give examples for human activities which they classify within a surveillance application stating "..human activities, such as {*Active, Inactive, Walking, Running, Fighting*..", and Nascimento et al. [7] give the examples of "entering or exiting a shop, passing, or browsing in front of shop windows" also for a surveillance context in a shopping center.

Zhao and Nevatia [8] set up a human locomotion model with the states *run, walk, stand* within a Finite State Machine. This definition is targeted at the tracking application they propose also for surveillance purposes.

A more general definition of human activity is given in [9], saying: "The current set of tasks is the user's activity", while a task is defined as "The association of a current state and a goal state" of the world. It is mentioned that a human seldom pursues only one task, but rather has a set of pending tasks, always switching between them. The same authors give examples for a collaborative work environment (see [10]), where persons are assigned roles like *speaker, listener* for activity recognition. But still, the granularity of tasks is not defined or restricted.

Hongeng et al. [11] implicitly define an activity based on an event taxonomy. An activity is equivalent to a simple or complex event, which is compiled from a set of basic events like *approach*, *leave*, *take object*. Activities can also consist of events triggered by multiple agents or entities, like several people working together for reaching one common goal. This definition of activity is hierarchical, building meta-activities from simple ones, even including several actors in one activity. With the described event taxonomy, it is possible to add activities to the recognition system in a natural way. But again, no closed-form definition for human activity is given concerning possible contents or granularity.

From these examples, it is obvious that for recognition of human activities, the goal of the system has to be taken into account. E.g. for a surveillance context, possible activities include *walking, standing, running, conversing,* or *taking an object* and *fighting*. In an office environment, where the system has to provide help and assistance to the user, activities may include *talking on the phone*, or *working with computer*.

For robot companions, the activity set which needs to be recognizable is again different. This is a major topic of research between the Research Activities 2 and 3. As long as no comprehensive definition for human activities in HRI situations has evolved, the work described in this report follows the depicted example-based approaches. The human activities which are to be recognized by the system are defined by example, deriving necessary activities from the demonstration scenarios. Different to surveillance tasks, the requirements for Human-Robot-Interaction (HRI) are more focussed on interaction and manipulation scenarios. The second demand to the used activity set is simply to show the abilities and weaknesses of the developed system. So partly, activities have been chosen simply for their motion pattern.

To be able to select proper recognition and classification methods, it is necessary to constrain the possible complexity of occuring activities. Within RA2, the focus lies on short-range activities which can be detected either instantly or from a short time window based on a physical description of the world. Higher-level components then will then include this information in their reasoning and decision-making processes.

The final set of activities which need to be recognized is still a topic of research within the Cogniron project (see [12]). Within RA3 in Cogniron, the Key Experiment scenarios are thoroughly investigated to retrieve a set of activities that on the one hand occur in the given situations, and on the other hand are needed by the robot for the decision making process. Current example activities include: *Bow, handshake, sit, walk* or *wave*, among others.

A. Human activities and Context

Recognition and interpretation of human activities is closely related to the question of modelling and determination of the *context*. This is due to the fact that human activities can not be detached from the context without becoming ambiguous in interpretation. Also, information about the current context eases recognition: An activity model can help to predict and rate activities and observations, and also include expectations in the recognition phase.

A usable context definition has been given by Dey in [13], which will also be used within this work:

> "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

This definition has also been adopted by [10]. Looking carefully at this definition, the following can be derived:

- Context contains the relevant *physical setting* of the environment. This includes objects, persons, places and their relations (see also [10]).
- The definition does not exclude information about current *tasks* and*goals*, or more generally, the user's (or system's) *intention(s)*. In fact, the term "any information that can be used to characterize the situation" may include anything from the state of mind of the user, if it is relevant.
- The definition does not claim that the "information that can be used to characterize the situation" is (or even can be) grouped semantically. There is even no claim that any relation between the bits of information exists that together make up context.

From this definition, two ways of exploiting context for activity recognition can be deduced.

- Context information may be used to parameterize the classification system, resulting in an expectation (and thus model) based framework. The individual recognition procedures do not use context information explicitly, but based on current context, different recognition procedures are active. This approach is e.g. used by Crowley et al. in [10] and [14].
- Information from context is used directly as input for classification. This approach does not work expectation based, but uses relevant information just as another data source which is directed in to classifier. This approach is described e.g. by [11], claiming that certain body movements together with the fact that an object is missing afterwards lead to the recognition of "pick up object".

This report adopts the second approach, using parts of the context as input for activity classification. We distinguish between information about body movements of the user called *intrinsic features*, and information about the environment (like objects, places, conditions like temperature etc.) called *extrinsic features*. Following the context definition of Dey (see above), both feature sets together make up context.

IV. Proposed approach for activity recognition

As described in sec. II, our proposed approach for activity recognition consists of three main steps: A *human motion capture* system gathers data of the human configuration over time, resulting in trajectories for each modelled limb and joint of the human body in 3d. The motion capture system called *VooDoo* is described in detail in [15], [16]. From this information, for each activity which has to be recognized a set of *model intrinsic features* is derived. These features do not rely on temporal segmentation, but are generated continuously. In addition, a set of *extrinsic features* can be taken into account. These features must be generated by external modules by observation of the environment. The feature synthesis is described in sec. V, the evaluation and selection process in sec. VI and sec. VI. The *classification step* is performed by a simple Feed Forward Neural Network (FFNN) which processes the feature stream. This FFNN has been trained with manually segmented training examples.

The whole process is depicted in fig. 1.

V. Feature extraction

It is not possible to exactly measure the quality of a feature, and different classifiers may prefer different feature types. However, we can still define important properties which characterize a good feature for the purpose of activity recognition. A good feature should

- be characteristic for at least one activity,
- separate between different activities,
- be reproducible for an activity independent from individual and temporal variations.

Three types of features are used within this context: Raw model data, filtered model data, and extrinsic features.

1) Raw tracking data: Obviously, the *raw tracking data* consisting of the whole body configuration trajectory can be used as feature set. A human who observes a reconstructed motion from this stream is directly able to recognize the performed activity, so this should also be possible for an appropriate classifier. Raw data can also be combined to retrieve more sensible features: Computation of the TCP height with respect to head height can e.g. help much in seperation of different activities. The raw tracking data is also referred to as *primitive feature set*.

In general, however, much model data is higly irrelevant, and even with the best classifiers of today, results are highly dependent on the data used during training and recognition. It seems highly optimistic that a classifier will be able to recognize all the activities with a limited training data set. Experimental data supports this, as seen in the results section of this report.

2) Statistical analysis: A high number of statistical methods exists that can be used to extract sensible features from a data set. Our features set contains the following:

- *Covariance* between different different input data vectors,
- *Principial Component Analysis* to detect the most relevant features in the current set, or to detect major motion directions (e.g. motion planarity),
- *Frequency analysis* to detect periodic motions.

This reveals a major difference in feature properties: Features can either be *time-dependent* or *snapshots*. Frequency analysis of e.g. elbow motion must be determined using a time window, thus taking motion history into account. TCP height with respect to the head is independent from history. This illustrates why different classifiers may prefer different feature sets: A classifier which itself takes history into account does

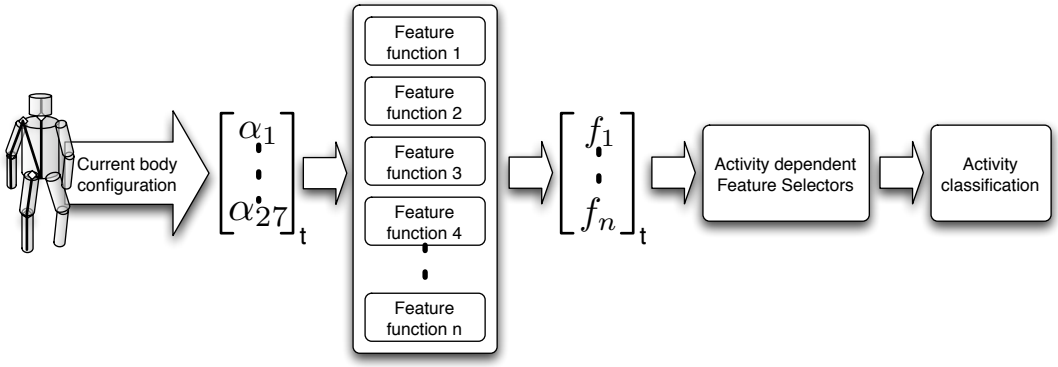


Fig. 1. Feature extraction and activity recognition process

not rely on time-dependent features as strong as a classifier without time modeling.

3) Extrinsic features: Many activities are very simple to recognize when not only the body motion of the human is given, but also information from the environment. This may comprise information about time of day, grasped objects, the current location, or other persons involved.

Generally speaking, this is context information. As the definition of context given in sec. III-A comprises *all* relevant information from the environment, only part of the context can (and must) be included for activity classification, because (a) only part of the world state can be measured, and (b) only part of all possible human activities is of interest. This included context knowledge is called *extrinsic features* in this report.

VI. Feature evaluation

Given a large feature set designed to capture a wide variety of activities, we should expect many features to be irrelevant for any given activity. In practice, these features will contribute nothing but noise to the classifier. By blindly feeding it all our extracted features, we depend on the classifier to be able to filter out irrelevant features. While it appears that feature selection does not improve predictability for most classifiers, it does improve training speed, and also seems to aid generalization (see [17]). The training data does not need to cover all possible variations any more: E.g. for training *waving*, the training set does not have to comprise sequences where the person stands, sits, and walks, only to train the *waving* movement for all other situations.

There are however other reasons to select a subset of features. By focusing on a few properties of the activity, it will be easier to provide stable recognition of activities even under heavy occlusion, as long as the features we need can be successfully extracted.

Likewise, having some insight into what is actually being evaluated removes a large part of the "black box" mystery from the activity recognition. This in turn may greatly simplify algorithm selection and parameter tuning, as well as provide a feedback loop between the person whose activities are being recognized.

Finally, performing activity classification with only a small set of selected features instead of using all of them significantly reduces the computational effort: A classifier that is fed 5 instead of 100 features is simply much faster.

To summarize, good feature selection has the following benefits:

- Removes irrelevant noisy signals.
- Allows faster training and better generalization.
- Removes redundant signals.
- Focuses recognition to a small set of properties.
- Tells us what is really being classified.
- Significantly reduces computational effort.

To determine a relevant subset of features for classification of a given activity, we first need to define the term *relevance*.

Mathematically, the most general statement we can make about a relevant feature F_i and a target class C is that for a given class value c and at least one feature value f_i, it satisfies the relation

$$p(c = C|f_i = F_i) \neq p(c = C) \tag{1}$$

which simply means that the posterior probability about the class of a sample after observing the feature is different compared to the prior.

Although the above definition seems intuitive, it is possible to find sets of clearly relevant features where it does not hold [17]. It seems necessary to define two relevance classes: *strong relevance* and *weak relevance*.

We say that feature F_i is strongly relevant if there exists a set S_i of all features except for F_i, i. e. $S_i = \{F_1, \ldots, F_{i-1}, F_{i+1}, \ldots, F_n\}$ and some value assignments f_i, c and s_i for which $p(F_i = f_i, S_i = s_i) > 0$ such that

$$p(C = c|F_i = f_i, S_i = s_i) \neq p(C = c|S_i = s_i). \tag{2}$$

We say that F_i is weakly relevant if it is not strongly relevant, and there exists a subset S'_i of S_i and some f_i, c and s'_i with $p(f_i = F_i, s'_i = S'_i) > 0$ such that

$$p(c = C|f_i = F_i, s'_i = S'_i) \neq p(c = C|s'_i = S'_i). \tag{3}$$

Strong relevance implies that a feature can not be removed without loss of prediction accuracy. Weak relevance implies that a feature can not always be removed without loss of prediction accuracy, depending on which subset of features it is used together with. In other words, weak relevance generally implies that a feature overlaps at least one other feature in the subset S_i.

While the definitions of equations 2 and 3 help us formalize the concept of feature relevance, they do not tell us how to find the probabilities they depend on. This is by no means a trivial problem, and much work has been done within this area. A well written introduction to the area is that of Guyon and Elisseeff [18].

In general, the best we can do is estimate the probabilities from sample data. Such methods are generally called *filters*. Several methods to do this have been proposed based on concepts from statistics and information theory.

Another approach is to forego the probabilistic analysis altogether, and instead look for empirically good features. This is the approach taken by the so-called *wrapper* methods. A wrapper method makes use of an embedded classifier to evaluate the relevance of a set of features. A comprehensive treatment of various wrapper methods is that of Kohavi and John [19].

The benefit of filter methods is that they are not affected by the properties of any classifier. This means that the evaluation could theoretically work equally good, or bad, for all classifiers. Filters are also generally orders of magnitude faster than wrappers. The main downside is that a filter can only give an estimation of the relevance.

Filter methods typically make use of various statistical means to evaluate the relevance of features. These are e.g. *Correlation Analysis* and the more general *Mutual Information Analysis*, sometimes also called *information gain*.

Mutual information is a statistical measure which loosely speaking attempts to measure the "amount of information" that is common between two data sets.

Mutual Information uses the concept of entropy to describe a measurement of the information that is common between two stochastic variables. The Mutual Information I between the stochastic variables X and Y is defined as:

$$I(X,Y) = H(X) - H(X|Y) \tag{4}$$

where $H(X|Y)$ is the conditional entropy of X given Y, that is, the remaining entropy if X given a fixed value of Y. If we substitute the definition of the entropy H we obtain

$$I(X,Y) = \sum_{x,y} P(X=i, Y=j) \log_2 \frac{P(X=i, Y=j)}{P(X=i) \cdot P(Y=j)} \tag{5}$$

VII. Feature selection strategies

We have now defined different ways to evaluate the relevance of features with respect to an activity. Based on this, features can be selected for each activity.

The point of feature selection is to find a minimum number of features to achieve the highest possible accuracy. These two goals are commonly mutually exclusive. We may however try to find a sufficiently small set of features that provide satisfactory prediction.

The problem of finding a minimum set of features is analogous to that of inference of minimal structures, which is known to be NP-hard in the general case. This leaves *exhaustive search* as the only method certain to find the optimum solution. A simplification of this is the *greedy selection*.

4) Exhaustive search: The simplest method is the exhaustive search, where we let an algorithm test all combinations of features in order to empirically determine the quality of recognition achieved using a given combination. Some quality value is recorded for a specific combination, and based on these values the relevance of every single feature and of every combination of features can be calculated.

As stated above, the exhaustive search is the only method known that is certain to find the true relevance of a feature. The downside, naturally, is the sheer amount of processing needed when working with large feature sets.

In order to test all combinations of k features out of a set of n we need $\binom{n}{k}$ evaluations, a number which is commonly too large for all but the smallest feature sets, especially for wrapper methods where each evaluation may take seconds or even minutes.

5) Greedy selection: One common form of lower order simplification is the greedy algorithm. A greedy algorithm has no fixed limitation on the maximum order considered, but the simplification lies in that we never let it re-evaluate former decisions, which leads to a significant reduction of the search space.

In terms of feature selection, this limit implies that a feature that was selected at one point, will remain selected forever. This is correct for strongly relevant features, but might be sub-optimal for weakly relevant features.

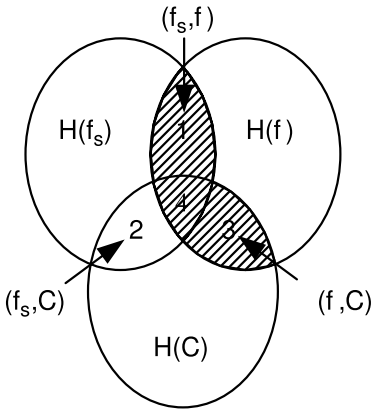
A. The Hill-Climbing Feature Selector

A very simple feature selection algorithm is the Hill-Climbing Feature Selector (HCFS) [20], also commonly called steepest ascent.

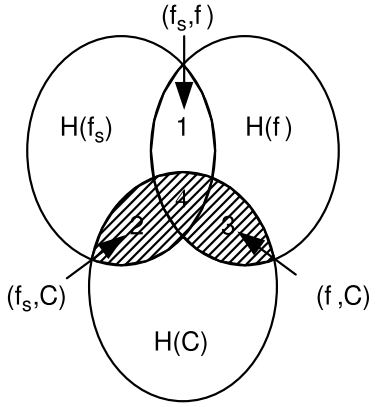
HCFS is a naive wrapper algorithm, which greedily forward-selects features starting from an empty set of selected features S. At every iteration we test all features v_i from the set of not selected features $V = \{v_1, ...v_m\}$ one at a time together with the set of selected features using the test algorithm $TEST$, typically a classifier.

The feature v that is found to add the most relevance to the set S as measured by $TEST$ is then added to this set, and we go on to select the next feature. The algorithm stops when we decide we have enough features, or when adding more features does not improve the relevance.

As with all greedy algorithms we tend to miss higher order relations between features, which means that while strongly relevant features will typically be selected while weakly relevant may be left out.



(a) MIFS



(b) Ideal

Fig. 2. MIFS (a) vs. Ideal Greedy Algorithm (b). The left region is the feature being evaluated, the right is the set of features already selected (if any), and the lower region is the target class.

B. Mutual Information Feature Selector

Battiti [21] reinterpreted the feature selection problem in terms of information theory:

> Given an initial set F with n features, find the subset $S \subset F$ with k features that minimizes $H(C|S)$, i.e. that maximizes the mutual information $I(C;S)$.

Using this interpretation, he proposed a greedy algorithm called *Mutual Information Feature Selector (MIFS)* to select features based on their mutual information with the target class $I(C;F_i)$.

The function $I(C;F_i)$ can be estimated through histogram generation over the sample data, i. e. we divide the sample data of C and F_i into small bins and calculate entropy and mutual information from the sizes of these bins using eq. 5.

Figure 2(b) shows in a information content venn diagram an ideal greedy mutual information based algorithm. Since the desired result is maximum predictability, the ideal algorithm would find features that maximize the overlapping between the feature set and the target class, i.e. the area $\{2,3,4\}$, or in mutual information terms: $I(C;F_i,F_s)$. Sadly, this turns out to be incredibly difficult to calculate using the histogram method.

The original MIFS makes use of a simplification, by calculating mutual information between every feature and all the features that have already been selected $I(F_i;F_s)$, and subtracting a fraction times this value from the final evaluation $R(F_i)$ of a feature.

$$R(F_i) = I(C;F_i) - \beta \cdot I(F_i;F_s) \tag{6}$$

Figure 2(a) shows the result of this equation, namely that instead of maximizing area $\{2,3,4\}$, we maximize $\{1,3,4\}$. This makes MIFS very sensitive to the amount of overlapping chosen, i. e. the value of β.

Kwak and Choi [22] suggested a simple way to approximate the desired area $\{2,3,4\}$. The shaded area for the ideal algorithm can be specified as $I(C;f_i,f_s)$, which can be rewritten as

$$I(C;f_i,f_s) = I(C;f_s) + I(C;f_i|f_s) \tag{7}$$

where $I(C;f_s)$ can be identified as areas 2 and 4, and $I(C;f_i|f_s)$ as area 3. Since $I(C;f_s)$ is constant for every feature evaluated, its value does not affect the relative effectiveness of a feature and we do not need to calculate this, which leaves $I(C;f_i|f_s)$ as the area to be maximized. This however turns out to be extremely hard in the general case. Kwak and Choi therefore suggest to expand this expression as

$$I(C;f_i|f_s) = I(C;f_i) - I(f_s;f_i) - I(f_s;f_i|C). \tag{8}$$

Assuming the information is distributed evenly throughout $H(f_s)$ in Figure 2(b), we may write

$$\frac{H(f_s|C)}{H(f_s)} = \frac{I(f_s;f_i|C)}{I(f_s;f_i)} \tag{9}$$

which combined with equations 8 gives

$$\begin{aligned} I(f_i;C|f_s) &= I(f_i;C) - \left(1 - \frac{H(f_s|C)}{H(f_s)}\right) I(f_s;f_i) \\ &= I(f_i;C) - \frac{I(f_s;C)}{H(f_s)} I(f_s;f_i) \end{aligned} \tag{10}$$

and finally we give an improved estimation $R_2(f_i)$ of the area $\{2,3,4\}$ as

$$R_2(f_i) = I(C;f_i) - \beta \left(\frac{I(f_s;C)}{H(f_s)} I(f_s;f_i) \right). \tag{11}$$

Once again, this only holds under the assumption that $H(F_s)$ is approximately evenly distributed.

Now, we have feature evaluation measures as well as selection strategies defined. The HCFS method uses a classifier to evaluate each feature (wrapper method) with sample data, while the MIFS method uses statistical relevance measures with sample the data sets to determine the feature set.

VIII. Activity classification

For the actual classification, a simple *Feed Forward Neural Network* was chosen. This decision has been taken for two reasons: First, a FFNN is an easy to use and simple classifier. It is good enough to prove or disprove the feasibility of the feature selection approach. After this first step, optimization can still be done using other classifiers which may be more suitable for the stated problem. The second questions concerns time-dependency. The FFNN is a time-independent classifier, which may cause problems and ambiguities when it is used for classification of time-dependent processes like in the given context. This problem is solved by including also time-dependent features like motion planarity, frequency analysis, etc. For details, see sec. V. So in this case, the time-dependency is dissolved in the feature extraction step, not during classification.

It is also possible to use a classifier which includes a time model, like HMMs, Bayesian Networks, or more sophisticated Neural Network approaches (e.g. Time Delay Neural Networks, TDNNs). This may result in better recognition rates; It also increases the effort of modeling and/or classifier design and training. Works on implementation and comparison of different classifiers are currently in progress.

For each activity which has to be recognized a solitary neural network has been used for the following reasons: As described in sec. VII, a different set of features is selected for each activity classification process. This means that the set of relevant features which for the input vector is different for each activity. Additionally, if only one network is trained for recognition of all activities, it has to be completely retrained when new activities are added. Also training data for one activity affects all NN weights, and thus the recognition rates for all activities.

For recognition of n activities, n neural networks exist. Each NN consists of 3 layers with

- k_i input neurons, with k_i the number of selected features for the given activity class c_i.
- 1 output neuron for the current estimation for activity class c_i.
- 10 neurons in the hidden layer. This has been chosen from experiments and experience. The number of selected features gives for all cases $k_i \leq 10$. Less than 10 neurons in the hidden layer decreased recognition results, while chosing more than 10 did not notably increase recognition rates.

IX. Experiments and results

In order to test and evaluate the proposed methods a series of experiments has been carried out. The goal was twofold:

- To test the efficiency of the feature extraction methods used when predicting the class, in other words the recognition result.
- To test the efficiency of the two feature selection algorithms and their influence on the recognition result.

The strategy used was to do two main experiments, called the *build-up* experiment and the *tear-down* experiment.

For the build-up experiment we start out with the tracking data as a primitive feature set and then build a large superset of features for classification using the methods developed in sec. V. This should give the best recognition results that can be achieved with the extracted features and the selected classifier.

Following this the tear-down experiment is done, where a small subset of highly relevant features are selected per activity, while trying to retain good recognition results.

A. Training data

A number of test activities had to be recorded. The activities were chosen with some considerations in mind. A good activity was decided to be:

- a natural activity that people commonly do.
- successfully and reproducibly tracked by the motion capture system.
- similar to at least one other activity, to show granularity of the system.
- dissimilar enough to other activitites to show the range of the system.

A set of 11 activities with these properties were selected and recorded. These activities were assigned an Id which we call *class*. A list of the activities can be found in table I.

TABLE I
Activities Used for Experiments

#	Name	Description
1	Balance	Balance on left leg
2	Bow	Upper body bow
3	Call	Right hand waving gesture towards body
4	Clap	Hand clap in front of body
5	Flap	Flap with both arms, “attempt to fly”
6	Handshake	Handshake with right hand
7	Kick	Kick with right leg
8	Manipulate	Object manipulation with both hands
9	Sit	Sit down on chair
10	Walk	Walk towards camera
11	Wave	Right hand waving

In order to have a sufficient data set for both training and testing 10 example sequences were recorded for one male and one female subject for each activity, making 20 sequences per activity. This gives in total 240 sequences with together 21222 frames.

The recorded sequences were manually segmented and assigned its correct activity. In the segmentation phase, frames were divided into three classes: Initialization, activity and finalization. Initialization frames were considered all frames up until the activity could be readily and correctly identified by a human. Activity frames were all frames where the activity was

clearly being performed. Finalization frames were considered frames where the activity was being halted.

For the training and evaluation, only the actual activity frames were used. The initialization and finalization frames generally tend to overlap the activity frames slightly. Therefore, frames have been classified as activity frames only when the activity was clearly identified by a human. This reduces the number of training frames, but increases the data quality.

The 240 segmented and classified sequences were then prepared into data sets suitable for training and classification. 50% of the recorded data were used for training, the other 50% for testing. To avoid *overfitting* in the NN classifier, the training data again has been divided into a training set and a validation set. Here we want the training set to be as large as possible, while having a validation set large enough to prevent overfitting. It was empirically determined that a validation set of 4 sequences and a test set of 6 sequences was appropriate.

For the training set we may then use at most 10 sequences, but in order to be able to test the recognition rate for varying train set sizes, five different training sets ranging from 2 to 10 sequences at intervals of 2 sequences were prepared.

The sequences were selected randomly from the total set with an equal number of sequences from the male and the female subject. To diminish the problems resulting from a random selection, five complete data sets with training, validation and test sequences were randomly created for each of the 5 training set sizes, making in total 25 sets.

B. Feature selection

The next step was to improve the recognition results by expanding the feature set. The methods developed in sec. V were applied in various ways to the tracking data. This was done in an iterative fashion, where a few new features were created and the recognition results were then observed. Further new features were added and once again the results were recorded.

In total this resulted in 118 intrinsic features, including the 45 in the primitive set. The complete list of features developed can be found in table II.

Used extrinsic feature is currently a *person holds object* binary feature which has been added to better recognize manipulation activities. As it is trivial to define an extrinsic feature which characterizes each activity (e.g. *second person holding one hand of the observed human* gives *handshake*) but which would be hard to detect in many cases, we have used only intrinsic features for the experiments and evaluation. These features can all be extracted from existing motion capture data and do not rely on any external modules. Furthermore, the feature evaluation and selection can then be rated from the results.

C. MIFS Parameter Selection

The MIFS algorithm takes one parameter, β, which determines the discounting of features based on their mutual information with already selected features. In order to find the optimal β for our purposes, different values were tested and

TABLE II
COMPLETE LIST OF INTRINSIC FEATURES

#	Description
1-3	Torso angle
4-6	Head angle
7-9	Left upper arm angle
10-12	Right upper arm angle
13-15	Left upper leg angle
16-18	Right upper leg angle
19-21	Left lower arm angle
22-24	Right lower arm angle
25-27	Left lower leg angle
28-30	Right lower leg angle
31-33	Head position
34-36	Right hand position
37-39	Left hand position
40-42	Right foot position
43-45	Left foot position
46-48	Torso angular velocity
49-51	Head angular velocity
52-54	Left upper arm angular velocity
55-57	Right upper arm angular velocity
58-60	Left upper leg angular velocity
61-63	Right upper leg angular velocity
64-66	Left lower arm angular velocity
67-69	Right lower arm angular velocity
70-72	Left lower leg angular velocity
73-75	Right lower leg angular velocity
76-78	Body velocity
79-81	Right hand velocity
82-84	Left hand velocity
85-87	Right foot velocity
88-90	Left foot velocity
91	Body position planarity
92	Right hand position planarity
93	Left hand position planarity
94	Right foot position planarity
95	Left foot position planarity
96	Right hand position variance
97	Left hand position variance
98	Hand covariance
99	Foot covariance
100	Hip covariance
101	Knee covariance
102	Hand distance period
103	Foot distance period
104	Right knee angle period
105	Left knee angle period
106	Right hip angle period
107	Left hip angle period
108	Right elbow angle period
109	Left elbow angle period
110	Right shoulder angle period
111	Left shoulder angle period
112	Distance between hands
113-115	Mean right hand position
116-118	Mean left hand position

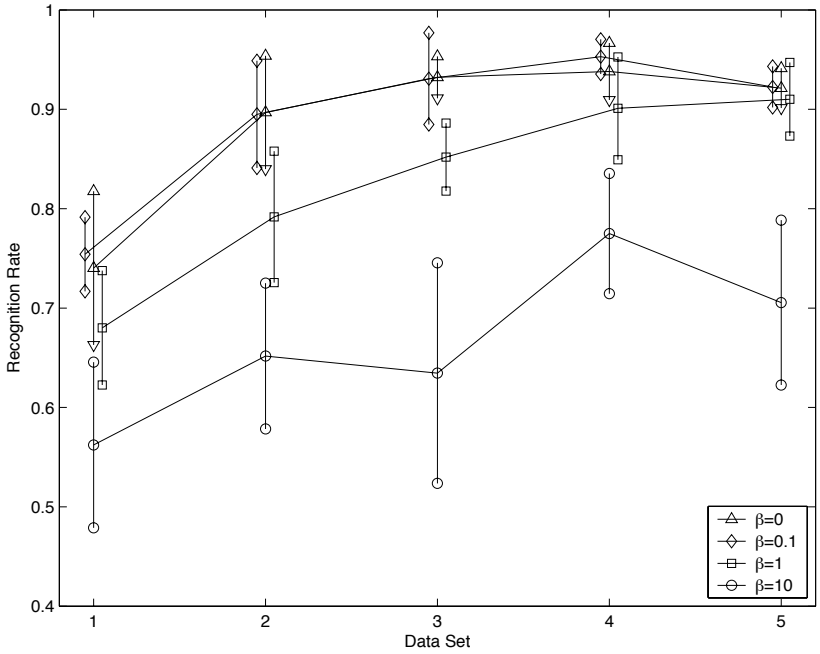


Fig. 3. Recognition Rates for MIFS Subsets With Varying β Value

the resulting recognition rate recorded, as β strongly depends on the actual data characteristics.

Figure 3 is a plot of recognition result for the various data sets, with standard deviation shown as error bars. From the plot we immediately see that lower values of β show strong results for the smaller data sets.

The primary effect of β is that for larger values the feature set selected will generally be smaller, since we stop the algorithm once new features do not add information according to our estimation. This is the reason why for the largest value of β we get extremely poor results, as the subsets selected contained at most two features.

As we increase the data sets the difference becomes smaller, and for our largest data set the results approach a common value, agreeing with the conclusions of Kwak and Choi [22].

Noticable is that the largest data set shows worse results than the second largest. This phenomenon is discussed in detail further on.

Based on these results the value chosen was $\beta = 0.1$ for all further MIFS runs.

D. Results and evaluation

There are two ways to measure how well an activity from the test set was recognized: The number of frames correctly classified, and the number of sequences classified.

While the number of frames is simple to count, it is harder to define exactly when a sequence has been recognized. As a vague definition we may say that the majority of the frames must be correctly classified in order to say that a sequence was recognized. As we shall see, with the high recognition rates achieved we can be certain that at least a significant percentage of each sequence was recognized.

Each frame was rated into one of four categories according to the firings of the neural networks:

- *Correct* when the neural network corresponding to the correct activity fired greater than 0.7.
- *Ambiguous* when at least one more neural network apart from the correct fired greater than 0.7.
- *Incorrect* when one or more neural networks corresponding to incorrect activities fired greater than 0.7, but the correct neural network did not.
- *Missed* when none of the neural networks fired greater than 0.7.

1) Build-up Experiment: Fig. 4 shows a bar diagram of the recognition results obtained per activity using only tracking data. Overall, the results turned out to be surprisingly good, with several activities showing reconition rates over 90 %. Especially noticeable are the activities *balance*, *bow*, *call*, *flap*, *sit* and *wave*, for which we obtain close to 100% recognition.

Some activities do not fare quite as well, especially *handshake* and *manipulate* where the recognition rate is around 50 % with intrinsic features only. Also *clap* and *kick* show less than 80 % recognition, and *walk* less than 70 %.

In order to explain why some activities were exceptionally well recognized we can take a look at which properties we can easily notice from tracking data itself. For the six activities that have over 90% correct classification we notice that there is one or more model angles that should readily tell the activity: *balance* (torso), *bow* (hip, torso), *call* (elbow), *flap* (shoulder), *sit* (hip) and *wave* (elbow).

Fig. 5 shows a bar diagram of the results obtained using the complete set of features extracted using the methods in sec. V.

Here we obtain nearly perfect recognition, with all activities

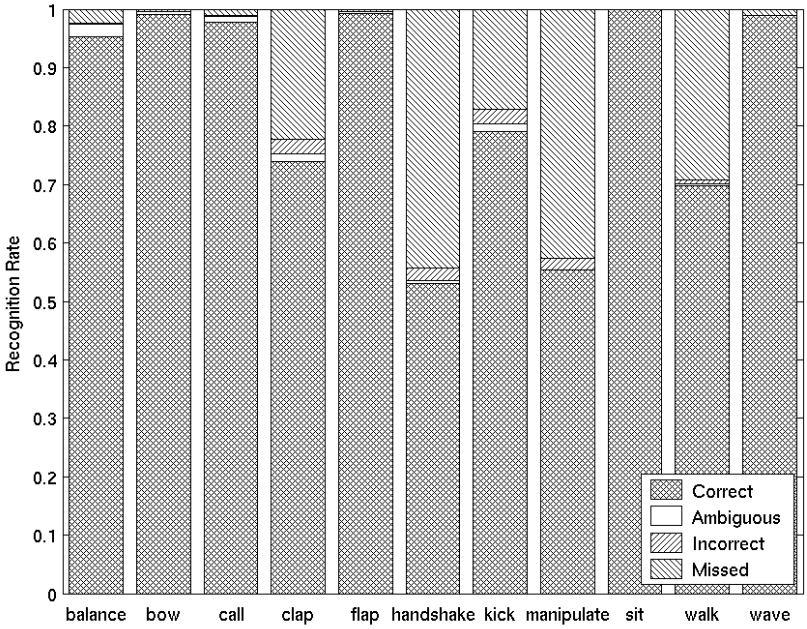


Fig. 4. Recognition Results Per Activity for the Primitive Feature Set

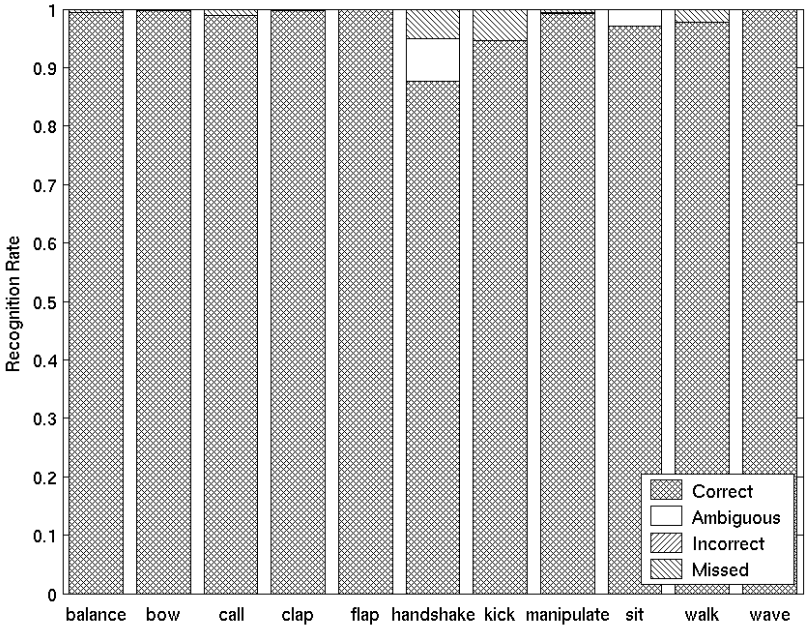


Fig. 5. Recognition Results Per Activity for the Complete Feature Set

TABLE III
AVERAGE RECOGNITION RATE OVER FIVE RUNS

Method	Correct	Ambiguous	Incorrect	Missed
Primitive	84.0 %	0.8 %	0.9 %	14.4 %
All features	98.3 %	0.6 %	0.0 %	1.1 %
MIFS 5 features	92.3 %	2.1 %	0.1 %	5.5 %
MIFS 10 features	95.7 %	1.6 %	0.0 %	2.7 %
HCFS 5 features	95.2 %	0.9 %	0.8 %	3.1 %

having more than 90% recognized frames. For the activity *handshake* we have some problems with ambiguity for approximately 5 % of the frames. Only *handshake* and *sit* now have more than 5% not recognized frames.

2) Tear-down experiment: The two selection algorithms, MIFS and HCFS, were run on the data sets. For both selectors a maximum of 5 features were chosen. This limit was placed mainly to try to confine the time needed for the HCFS algorithm, taking already more than 24 hours for the given computation. In the case of MIFS we could in practice let it choose all features until the mutual information is exhaused, but to make its results comparable to HCFS the same limit was applied.

Fig. 6 shows the recognition result per activity using the feature subset MIFS–5. The average recognition rate of 92.3 % places the MIFS–5 set in between the primitive and the complete feature sets.

The selected feature subset enables excellent recognition of many activities, above 95 % for eight of the 11 activities. However two activities, *call* and *handshake* show very poor results with only around 70 % recognition.

In order to see if a larger feature set would improve on these results, another MIFS run was done, this time producing the set MIFS–10, with 10 features. Fig. 7 shows the recognition result per activity using the feature subset MIFS–10. The average recognition rate of 95.6 % shows a slight improvement compared with MIFS–5.

We notice from the figure that *call* has reached above 90 %, but *handshake* is still only around 80 %.

It seems hard to justify adding five new features for only 3.3 % improvement.

Fig. 8 shows the recognition results per activity using the feature sets selected by HCFS. The average recognition rate of x % places the HCFS–5 set in between the MIFS–10 and the complete feature sets.

Table III shows the recognition rates of all presented experimental setups as an average over 5 runs from the discussed Monte-Carlo simulation.

E. Learning speed analysis

One measurement is how well the system generalizes, that is, how well the classifier is able to learn a concept instead of just the very examples used in training. As was stated in sec. VII, one of the desired properties of feature selection is quicker generalization. In other words, we should be able to train the system with fewer examples.

In order to test this a series of tests was carried out with increasing number of training sequences. For each activity training was done with 2, 4, 6, 8 and 10 sequences, after selecing features with MIFS and HCFS. Additionally, 4 validation sequences and 6 test sequences were used. All sequences were selected randomly from the 20 recorded. Just like above, to alleviate the problems with randomization, five sets of each size were created.

Figure 9 is a plot of average recognition rate versus number of training sequences for the primitive features, the entire set of features, and for subsets selected by MIFS and HCFS.

As can be seen the complete set of features is constantly superior to the other sets. Already with one training sequence the recognition rate is above 90%. With this small set we have a noticeable region with ambiguous results and even a few percent incorrect classification. As we add more training sequences we essentially eliminate everything but the correct and missed sets.

The primitive set is able to match the subsets selected by our two algorithms with the smallest training set for the recognition rate. Using more training data improves recognition very slowly hoewever. Additionally, both the miss rate and the error rate are considerably higher than for the three other sets.

The two selection algorithms perform similarly at the beginning. While HCFS has a slight lead with the smallest set, MIFS actually surpasses at two training sequences per activity, where the incorrect classifications have been nearly eliminated. At this point however, MIFS has already reached its maximum and actually performs slightly worse with 3 training sets. HCFS on the other hand improves nearly linearly with the number of sequences used, closely approximating the results achieved with the entire set.

F. Separation of strong and weak features

Keeping in mind the distinction between strong and weak relevance it is interesting to see if this distinction holds up in our test results. The problem here is naturally to separate the two classes. One way to define a strong feature in terms of the results obtained is as a feature that is invariably selected by the algorithms.

Fig. 10 is a breakdown of the features selected by the MIFS algorithm. For each activity the features are listed on the horizontal axis, and the vertical axis shows the number of times it was selected out of the five test sets. Additionally, the patterns show the different set compositions, so that all blocks having the same pattern were selected together.

Similarly, fig. 11 is a breakdown of the features selected by the HCFS algorithm. Features that occur in both selections and in each run can definitly be considered as strong features. All other features may be considered as weak, being only relevant together with others. It should be noted that this is not necessarily true, they may also turn out to be strong features as well; limiting the feature set to a maximum of 5 also influences feature evaluation.

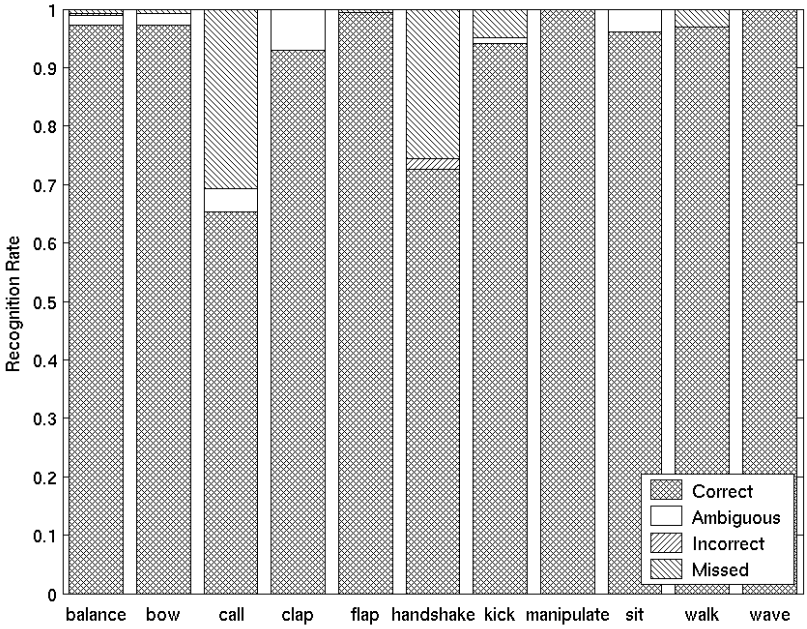


Fig. 6. Recognition Results Per Activity for the MIFS–5 Feature Set

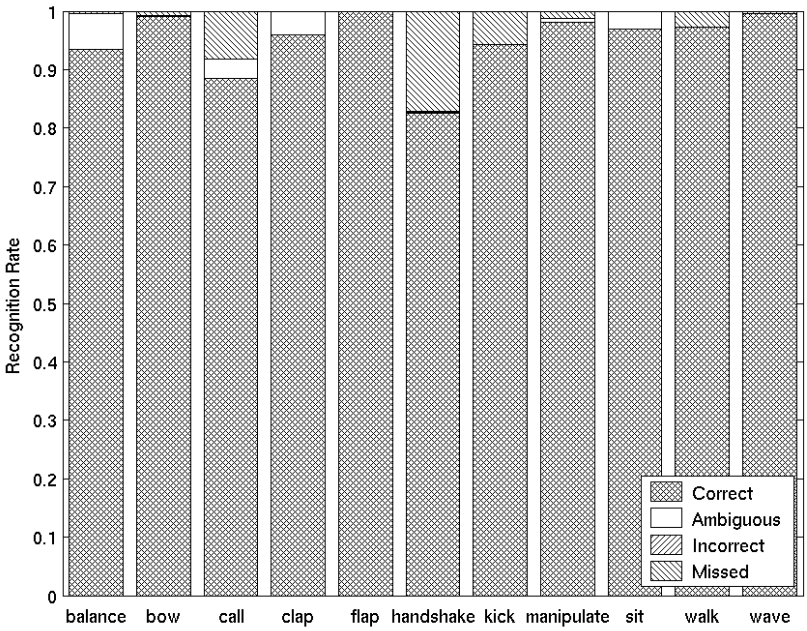


Fig. 7. Recognition Results Per Activity for the MIFS–10 Feature Set

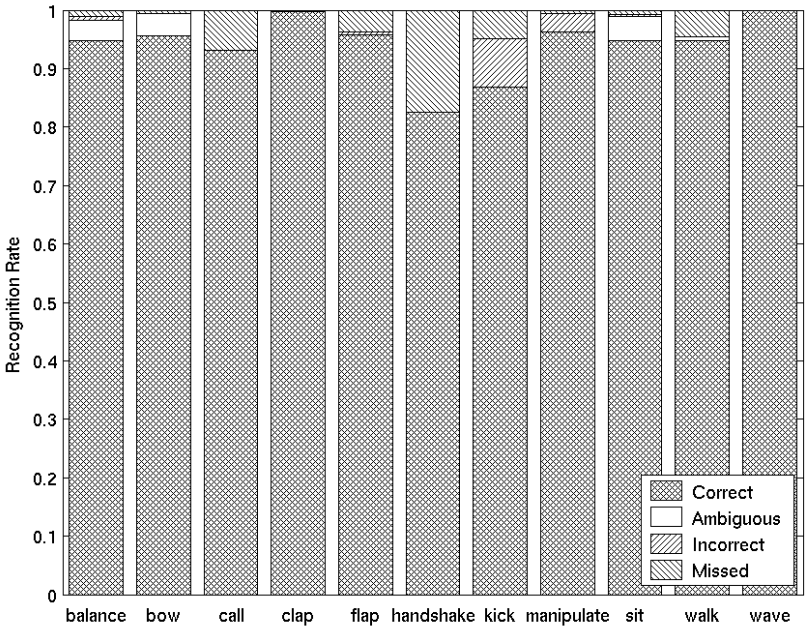


Fig. 8. Recognition Results Per Activity for the HCFS–5 Feature Set

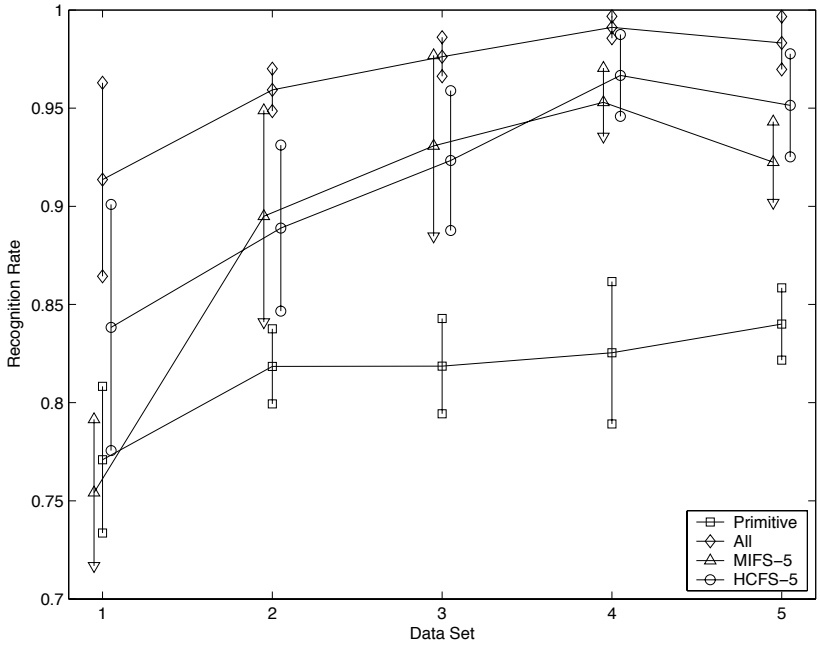


Fig. 9. Learning speed analysis for different feature sets and selections

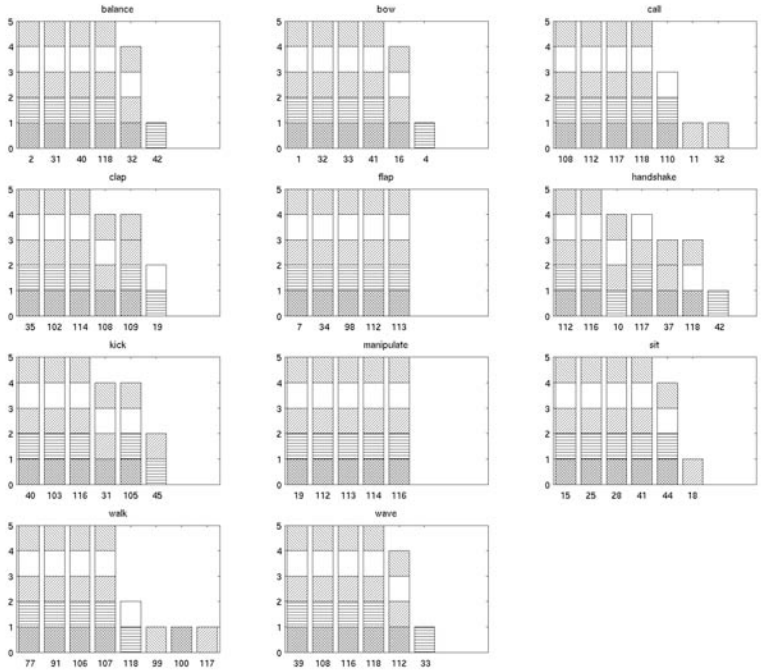


Fig. 10. MIFS Feature Subset Breakdown

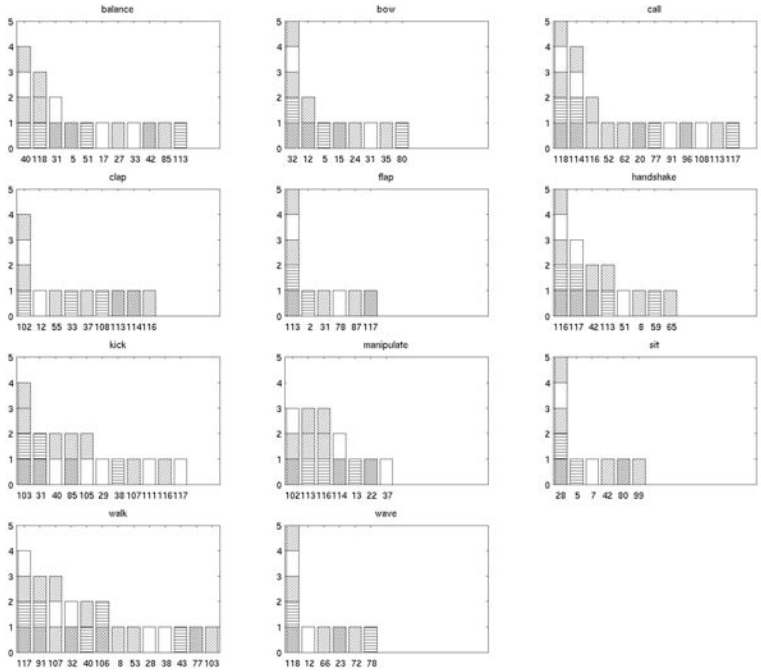


Fig. 11. HCFS Feature Subset Breakdown

X. Summary and conclusions

This report has focused on the problems of extraction, evaluation and selection of features to recognize human activities using an inference engine.

Features were extracted using body model data, various statistical methods as well as frequency analysis. Evaluation was done using mutual information analysis, and selection based on this analysis was compared with manual selection.

The selected features were then used to train and query a neural network and the recognition rate was used as an evaluation of the system as a whole.

This work has shown how a combination of feature extraction, evaluation and selection can work together to provide a high quality data set for use with an inference engine.

Feature extraction is not necessarily a difficult problem. In fact, as the feature evaluation step has shown, many of the best features turn out to be model data or very simple functions of it. For some activities, more complex features are needed.

Feature selection can be done using a variety of method. A few such were attempted in this work, and the idea behind selection turns out to be sound.

To conclude, we may draw the following conclusions:

- Feature extraction is effective and relatively simple.
- Feature evaluation by statistical means is working.
- Feature selection is sensible and usable.
- Extraction, evaluation and selection work well together.

Especially notable is how well the combination of feature extraction and selection works. With careful selection, even simple features may contribute significantly to recognition results.

XI. Future work

The feature extraction step can be expanded almost without limits. It is certainly possible to apply more advanced statistical means to extract more complex features. However, it should be kept in mind that the evaluation tends to suggest that complex features are not generally exceptionally good.

Thus, the main effort will be put on the improvement of the classificator. Usage of FFNNs has proved to be effective, but other classifiers may perform even better. Classifiers with and without intrinsic time model will be implemented and evaluated, such as HMM, Bayesian Networks and SVM.

Extrinsic features will be incorporated in addition to the mentioned features in sec. IX-B which has been added manually for the described work. These have to be selected carefully, as there also has to be a recognition module for each of them. Obviously, special extrinsic features considerably simplify recognition of the associated activities.

References

[1] C. Cédras and M. Shah, "Motion-based recognition: a survey," *Image and Vision Computing*, vol. 13, no. 2, pp. 129–155, March 1995.

[2] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, January 1999.

[3] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, March 1999.

[4] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Reconition*, vol. 36, no. 3, pp. 585–601, 2003.

[5] P. C. Ribeiro and J. Santos-Victor, "Human activity recognition from video: Modeling, feature selection and classification architecture," in *Proceedings of the International Workshop on Human Activity Recognition and Modelling 2005*, vol. 1, 2005, pp. 61–78.

[6] F. Pla, P. Ribeiro, J. Santos-Victor, and A. Bernardino, "Extracting motion features for visual human activity representation," in *Pattern Recognition and Image Analysis: Second Iberian Conference, IbPRIA 2005, Estoril, Portugal, June 7-9, 2005, Proceedings, Part I*, J. S. Marques, N. P. de la Blanca, and P. Pina, Eds., vol. 3522. Springer-Verlag GmbH, 2005.

[7] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, "Segmentation and classification of human activities," in *Proceedings of HAREM International Workshop on Human Activity Recognition and Modelling*, 2005.

[8] T. Zhao and R. Nevatia, "Ttracking multiple humans in complex situations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, September 2004.

[9] J. L. Crowley, "Context Aware Observation of Human Activity," in *Proceedings of ICME 02*, 2002.

[10] J. L. Crowley, "Context Aware Observation of Human Activity," in *PSIPS 2004, Oulu, Finland*, June 2004.

[11] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods," *Computer Vision and Image Understanding*, 2004.

[12] N. Otero, C. Nehaniv, K. Dautenhahn, J. Saunders, and A. Alissandrakis, "Naturally occuring gestures in a human-robot teaching scenario: An exploratory study," School of Computer Science, Adaptive Systems Research Group, University of Hertfordshire, College Lane, Hatfield, Hertfordshire, AL10 9AB, UK, Tech. Rep., 2005.

[13] A. K. Dey, "Understanding and using context," in *Proceedings of Personal and Ubiquitous Computing*, 2001.

[14] J. L. Crowley and P. Reignier, "Dynamic composition of process federations for context aware perception of human activity," in *Proceedings of KIMAS 03*, 2003.

[15] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3d human body tracking with an articulated 3d body model," in *submitted for ICRA06*, 2005.

[16] S. Knoop, S. Vacek and R. Dillmann, "Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP," in *Proceedings of the International Conference on Humanoid Robots (Humanoids 2005)*. Tsukuba, Japan: IEEE-RAS, 2005.

[17] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 121–129, 1994.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 1157–1182, 2003.

[19] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[20] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proceedings of the 13th International Conference on Machine Learning*, vol. 1, 1996, pp. 284–292.

[21] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, July 1994.

[22] N. Kwak and C.-H. Choi, "Improved mutual information feature selecor for neural networks in supervised learning," in *Proceedings of the International Joint Conference on Neural Networks, 1999*, vol. 2, 1999, pp. 1313–1318.

A Multi-Modal Object Attention System for a Mobile Robot*

A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer
Faculty of Technology
Bielefeld University
33594 Bielefeld, Germany
ahaasch@TechFak.Uni-Bielefeld.DE

***Abstract*— Robot companions are intended for operation in private homes with naive users. For this purpose, they need to be endowed with natural interaction capabilities. Additionally, such robots will need to be taught unknown objects that are present in private homes. We present a multi-modal object attention system that is able to identify objects referenced by the user with gestures and verbal instructions. The proposed system can detect known and unknown objects and stores newly acquired object information in a scene model for later retrieval. This way, the growing knowledge base of the robot companion improves the interaction quality as the robot can more easily focus its attention on objects it has been taught previously.**

***Index Terms*— object attention, human-robot interaction, robot companion**

I. Introduction

Developing *robot companions* that support a natural interaction with a human user is a challenging research topic. The basic idea of a robot companion is that it is used as a personal robot which a user shares his private home with. Thus, the interaction interface has to match all requirements for an easy usability, so that even naive users are able to interact with the robot without an extensive training phase. Furthermore, since robot companions are meant to be used for several tasks (e.g., assistance, entertainment, fetch-and-carry, etc.) it is essential that they are able to interact with their environment autonomously. However, before being able to act autonomously, a robot companion must learn about its environment.

As robot companions are intended primarily for private homes, a typical application focusing on the learning aspects is the so-called *home tour scenario*. The core idea of this scenario is that after the user has bought a new robot companion in a store, he takes it to his home and shows the robot all locations and objects that are relevant for later use. As a basis for human-robot interaction, the robot has to perceive potential communication partners in its vicinity and focus its attention on them. However, the robot must not only be able to detect potential communication partners, but also objects and locations that an actual communication partner is referring to. In accordance to the transfer of knowledge between humans this means that the robot must react on the user's actions, like, e.g., deictic gestures and verbal utterances. From the cognitive perspective these different modalities are combined in the capability of *joint attention*. From the robotics perspective, joint attention describes the process that enables a robot to look at the object which the user is referring to. When the robot companion is able to focus on the referenced object, learning new objects and updating information about objects already learned becomes a central capability necessary to perform autonomously later on. Additionally, a robot companion has to have a specific knowledge base in order to not only store, but also efficiently access all information gathered during interaction.

In this paper we concentrate on the aspect of focusing the robot's attention on objects. Focusing of attention and learning of objects has been demonstrated in static setups (e.g., [1]), but performing this task on a mobile robot is even more challenging. Our approach is realized by a methodology that we call *object attention system*. Its task is to focus the robot's attention on objects that become of interest because the user is referring to them using multiple modalities.

For focusing the robot's attention on objects, it is necessary to distinguish between known and unknown objects since it cannot be assumed that every possible object in a private home is known by the robot a priori. Especially learning unknown objects is a difficult task because the corresponding recognition methods are usually view-based and, therefore, depend mostly on visual object features. In order to face this challenge we place emphasis on the processing of multi-modal input by incorporating gesture information and verbally specified object properties to focus the robot's attention on the correct object. Additionally, in contrast to the above mentioned joint attention, we do not only let the robot look on the same object which the user refers to, but also collect all information available about this object. Through storing the multi-modal object properties in a knowledge base, a learned object can be easier detected during later interactions.

The remainder of this paper is organized as follows: In Section II we will discuss earlier work related to our approach. Section III provides a brief overview of the robot architecture used on our mobile robot platform BIRON. Section IV explains the gesture recognition methodology we

*This work has been partially supported by the European Union within the 'Cognitive Robot Companion' (COGNIRON) project (FP6-IST-002020) and by the German Research Foundation within the Collaborative Research Center 'Situated Artificial Communicators' as well as the Graduate Program 'Task Oriented Communication'.

In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2005. IEEE.

© IEEE. Personal use of this material is permitted.However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: IEEE Intellectual Property Rights Office / IEEE Service Center / 445 Hoes Lane / Piscataway, NJ 08855-1331 / phone: (732) 562-3966 / fax: (732) 981-8062

are following to detect user gestures. Details of the object attention system integrating gesture recognition results are described in Section V. Finally, we present exemplary results obtained with our attention system in Section VI before the paper ends with a summary.

II. Related Work

An important aspect for a natural human-robot interaction is the ability of a robot to focus its attention on the objects referenced by the human. In human-human interaction, a variety of cues are applied to focus on the same object. A detailed discussion of this joint attention for use on robots can be found in [2]. However, an additional problem for a robot companion is the fact that the object of interest may be difficult to detect and, therefore, additional cues need to be incorporated in a technical system. An obvious source of information are deictic gestures and verbal descriptions of the object in question. Different systems are described in the literature that apply such additional cues to detect the object that is in the focus of the human-robot interaction.

The robot Leonardo [3] is capable of detecting the interaction of a human with saliently colored buttons arranged around the stationary robot. Leonardo recognizes deictic gestures in combination with speech. Specifically, it is possible to label buttons by giving verbal information as well as to teach the robot how to use these buttons. The robot's environment is equipped with an additional camera that views the scene from above to solve this task.

Another static setup for analyzing interactions between objects and the user's hand is described by Utsumi et al. [4]. Here, no previous knowledge about the objects is necessary since an appearance-based approach is used for modeling objects. However, in contrast to Leonardo's setup with only one camera mounted at the ceiling, this approach uses several cameras to observe the scene.

An impressive system running on a mobile robot is presented by Ghidary et al. [5]. The robot is able to learn objects by analyzing speech commands and hand postures of the user. The user gives verbal information about the object's size and can describe the spatial relations between objects e.g. by phrases like 'left of my hand'. The rectangular views of the objects learned are stored in a map representing the robot's environment and can be used for later interactions. Although the interaction system is very limited and the resulting map is rather coarse, this system can be compared to our approach as it also builds up a long-term memory about objects in the environment. However, we focus on a more detailed representation of objects and their later recognition in order to support natural human-robot interaction going beyond simple navigational tasks.

III. Overall system architecture

The approach outlined in this paper is developed and implemented on our mobile robot BIRON [6]. As BIRON is intended to be used as a robot companion, it is equipped with an architecture that on the one hand is capable of managing the different modules processing various input modalities. On the other hand it controls the hardware. To accomplish these tasks, we developed a three-layer architecture [7], as it is the most flexible way to organize a system which integrates autonomous control and HRI capabilities (see. Fig. 1).

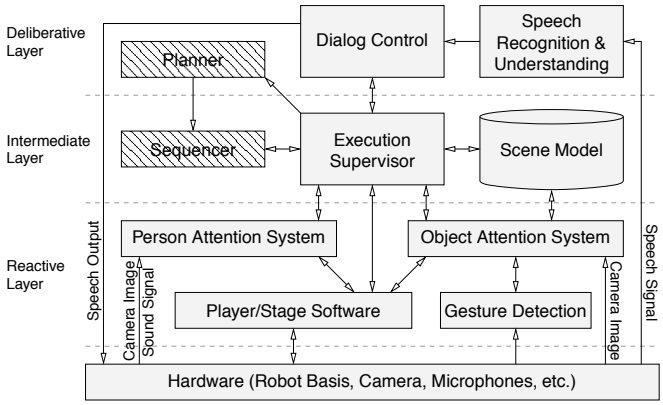


Fig. 1. Overview of the BIRON architecture (optional modules currently not implemented are drawn hatched).

A central execution supervisor is responsible for controlling the communication between the modules in our architecture. We use XML as data format and a specially developed XML-based communication framework for implementing different communication patterns. The combination of the communication framework with our architectural methodology results in the System Infrastructure for Robot Companion Learning and Evolution (SIRCLE) [8]. Next, we briefly describe the individual components of our architecture.

The dialog control component is part of the deliberative layer. It manages dialogs and receives instructions from the interaction partner via the automatic speech understanding system which is located in the same layer. The dialog module sends valid instructions to the execution supervisor which is located in the intermediate layer and routes messages between modules. The scene model is also located here and is responsible for storing information about objects provided by the object attention system. Note that the system can be complemented by a path planner and a sequencer, but as we focus on HRI these components are currently not integrated.

The object attention system located in the reactive layer is described in Section V. It receives information about pointing gestures from the gesture detection outlined in Section IV. Furthermore, the person attention system [9] is located in the reactive layer. It detects communication partners among persons present in the vicinity of the robot. The robot's hardware is controlled by the *Player/Stage* software [10] providing an interface to the robot's sensors and actuators. This enables us to easily replace the controller by a more complex one which may also include, e.g., obstacle avoidance.

IV. Recognition of Pointing Gestures

A robot companion should enable users to engage in an interaction as intuitive as possible. Deictic gestures are an essential part of human-human communication, therefore they are used for the presented human-machine interaction. Such gestures are usually performed by humans to reference an object in the vicinity of the hand.

One task in the presented scenario is to learn objects, so object recognition results are not always available for each object the human points at. For the motion based activity recognition this results in two cases: (a) An unknown object is referenced or (b) the human is pointing at an object previously known to the system. For the first case a probabilistic approach is used to detect pointing gestures based on motion of the hand represented by its trajectory. To cope in the second case with pointing gestures to known objects, both, the hand trajectory and symbolic data describing the objects in the scene are used to detect a deictic gesture referencing a specific object. This combination is necessary if several objects are present in the scene as the direction of the moving hand has to be considered.

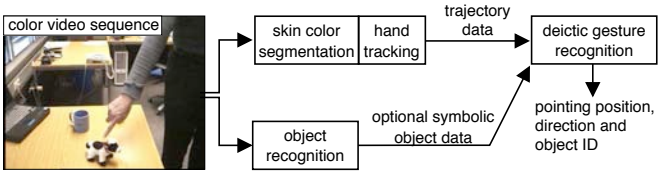


Fig. 2. Modules for the deictic gesture recognition.

The automatic detection of pointing gestures is based on visual input using different modules shown in Fig. 2. Information about the hand trajectory needed for the motion analysis is obtained by applying a skin color segmentation and tracking the resulting regions over time using a Kalman filter with a constant acceleration model. This trajectory data is used in the deictic gesture recognition module where the symbolic data from object recognition is incorporated. The results of this module are the hand position, its pointing direction, and optionally the referenced object.

In the following the two cases for the recognition of pointing gestures are explained in detail:

A. Pointing at previously unknown objects

In [11] Black and Jepson describe their extension of the CONDENSATION algorithm by Isard and Blake [12] to realize a Condensation-based Trajectory Recognition system (CTR). We apply this method to analyze the motion of the hands. This Particle Filtering algorithm compares several models $_i$ – each describing an activity – with the observed trajectory data $\mathbf{z}$ of the hand using a sample set. The temporal characteristics of a motion is included by using the data of several previous time steps. The models $_i$ are scaled in time and amplitude. The quality of the match of such a scaled model – the sample n with its parameter vector $\mathbf{s}^n$ – and the measurement $\mathbf{z}$ is expressed in a weight $^{(n)}_t$.

Recognition of activities is performed by calculating an end probability $p_{end}(\ _i)$ for each model by summing the weights $^{(n)}_t$ of all samples with a matching position above 90% of the trajectory length. Recognizing a model is complete, when the threshold for the end probability $p_{end}(\ _i)$ for one model is reached.

The characteristics of pointing gestures are their velocity Δr and change of direction Δ of the hand. In contrast to Black and Jepson's work, we use this representation and not the x and y velocities for the trajectory and the models $_i$. In this way, we are able to abstract from the absolute direction of the gesture.

B. Pointing at known objects

During a pointing gesture the referenced object can be expected in a certain spatial area relative to the approaching hand: The *context area*. The symbolic information of known objects is integrated in the existing CTR process using this spatial context. The approach is outlined below, for details see [13].

In order to have a variable context area we extend the vectors of the models $_i$ by adding parameters describing this area. It is defined as a circle segment with a search radius c_r and a direction range, limited by a start and end angle (c_α, c_β) relative to the direction of the hand tracked (see Fig. 3).

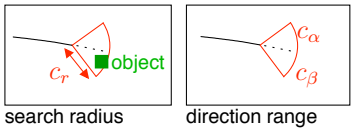


Fig. 3. Parameters for the definition of the context area.

In every time step the context area is checked for each sample $\mathbf{s}^n_t$. If there is more than one object in the context area of the sample at the current time index, one object is selected at random. For adding this symbolic data to the samples of the CTR algorithm we extend the sample vector $\mathbf{s}^n$ by a parameter ID_t denoting a binding to a specific object: Once the sample contains an object ID this will be propagated with the sample.

Additionally, we extend the calculation of the sample weights from [11] by a multiplicative *context factor* P_{symb} representing how good the bound object fits the expected spatial context of the model:

$$^{*(i)}_t \propto p(\mathbf{z}_t|\mathbf{s}^{(i)}_t)\ P_{symb}(\mathrm{ID}_t|\mathbf{s}^{(i)}_t) \qquad (1)$$

The influence of the parameter P_{symb} on the recognition results is described in detail in [13], here we use $P_{symb} := 1.0$ if the expected object is present and a smaller value,

e.g., $P_{symb} := 0.6$ if no object is bound. This leads to smaller weights $^{*(i)}_{t}$ of samples with a missing context so that these samples are selected and propagated less often by the CTR algorithm.

On recognition of an activity the parameter IDs in the sample vectors are used for evaluating the object the human pointed at. For each object O_j the sum p^O_j of the weights of all samples belonging to the recognized model $_i$ that were bound to this object are calculated. If the highest value is larger than a defined percentage ($T_O := 30\%$) of the model's end probability $p_{end}(\ _i)$, the object O_j is selected as being the object the human pointed at.

Hence, the benefit of the approach described is a robust recognition of deictic gestures including the position in the image and the pointing direction. The system is also able to detect hands pointing at previously known objects.

V. Object Attention System

The object attention system (OAS) is activated on demand when the user is referring to an object or the robot has to interact with an object autonomously. In order to retrieve visual information about objects, BIRON uses an active camera. Additionally, a stereo camera is used to determine the object's position relatively to the robot as described in [14]. Normally, camera lenses exhibit a limited field of view with regard to their opening angle which might make it necessary to reorient the camera to relevant parts of the current scene. In the context of the home tour scenario outlined above the scene area to which the user refers to is considered important. After the robot has focused its attention on this region of interest (ROI), the acquisition of object information, like position and view, is completed. Next, the object data collected has to be added to the robot's knowledge base. Besides storing this information, the knowledge base must also allow retrieving stored object information and updating already stored data. Furthermore, additional information given verbally by the user is stored as well.

The coordination of verbal information, gestures, and salient object features (e.g., color, shape, etc.) perceived by the camera, as well as control of hardware components like the camera and the robot basis inside the OAS (see Fig. 4) is realized by a finite state machine (FSM). Input is provided by the camera, the dialog module, the automatic speech understanding component (ASU), and the gesture recognition module.

Since it cannot be assumed that every object is known to the robot a priori, a distinction between known and unknown objects has to be made. To determine whether an object is already known, the robot's *scene model* is used. The scene model is based on the concept of an active memory [15] and provides the OAS not only with features about objects already stored in it, but also with appropriate image patterns for the object recognizer used. In addition, information that is given

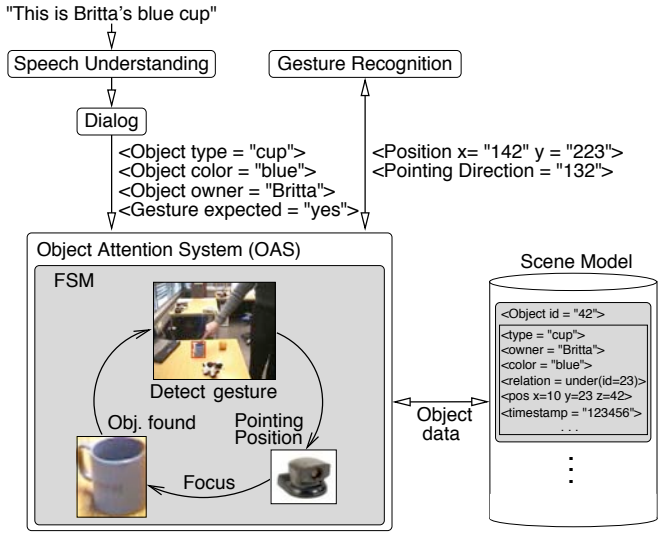


Fig. 4. Processing chain of the object attention system for given verbal input information.

verbally by the user is stored as well. However, the scene model for a robot companion must also provide additional capabilities. Since the robot's environment is continously changing, the scene model has not only to be able to *forget* obsolete information about objects (e.g., the position of a cup three months ago), but also to update this kind of information if required.

The OAS is activated if a phrase which contains the description of an object like "This is Britta's blue cup" is addressed to the robot. At first, the FSM (see Fig. 5) is in the idle state, called *Object Alertness* (ObjAlert). If the OAS is provided with data by the dialog module (see Fig. 4), the FSM changes to the *Input Analysis* (IA) state. Depending on a lexical cue like, e.g., "this" or "that", the ASU determines that a gesture is expected. As a consequence, the gesture recognition module is activated. This module supplies the OAS with the user's hand coordinates and the direction of the corresponding pointing gesture. Thus, an area within the camera image is selected as resulting ROI. In case the dialog module sends the description of the object (e.g., type, color, owner, etc.) to the OAS, an inquiry to the scene model is initiated to check whether the object type that is sent by the dialog component is already known or not. Next, we describe the processing in case the object type is known to the robot and afterwards if it is unknown.

A. Previously known objects

The search for a known object type involves an object detection process. For this task we use an object recognizer that is trained only for a few objects right now. It is based on the fast Normalized Cross-Correlation (NCC) algorithm described in [16]. Other approaches for recognizing objects (e.g., [17], [18]) can be integrated in the system as well.

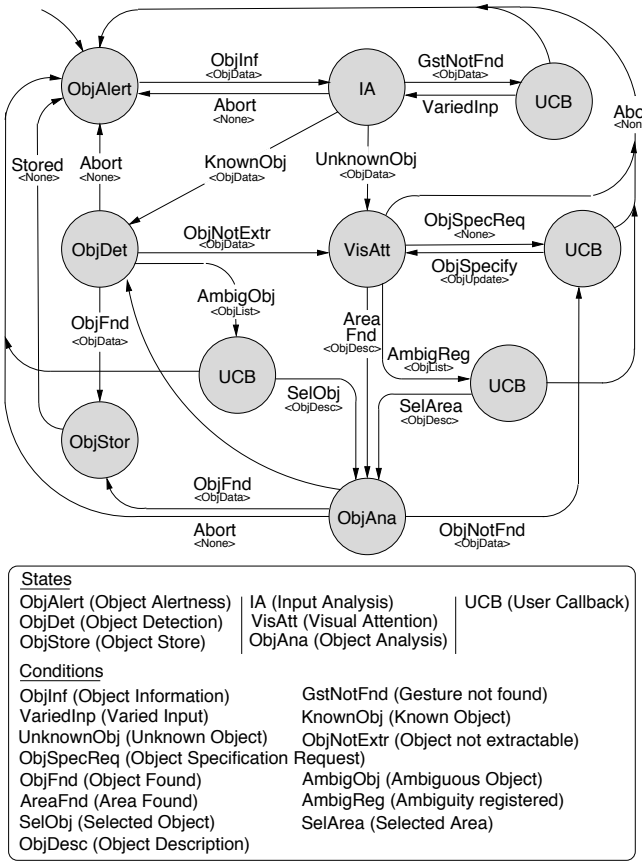


Fig. 5. The finite state machine of the object attention system.

However, as we focus on the interactive object learning and not on the issue of object recognition, a simple NCC algorithm is sufficient.

After retrieving an appropriate image pattern for the known object, the FSM switches from the IA state to the *Object Detection* (ObjDet) state. Within the ObjDet state the OAS uses the image patterns (e.g., for cups) in order to apply them with the object recognizer. Additionally, the camera is reoriented based on the hand coordinates and the pointing direction that are provided by the gesture recognition module. In addition to the reorientation of the camera, the position of the hand is used to determine the relevant ROI. Now, the object detection process is initiated and on finding an object a confirmation message is sent to the dialog module and the FSM switches to the *Object Store* (ObjStor) state. In this state the position of the object obtained is stored in the scene model. Finally, the FSM returns to the ObjAlert state and the OAS awaits new orders.

If two or more objects of the same type are found during the detection phase in the ObjDet state, the FSM switches to the *User callback* (UCB) state. Next, a message is sent to the dialog component to find out which specific object is meant by the user. Consequently, this helps to resolve ambiguities. After the dialog module receives a more detailed description, like “The left one.”, the FSM switches to the *Object Analysis* (ObjAna) state. In this state a new ROI is determined based on the information from the gesture detection and the lexical cue “left”. Now the FSM returns to the ObjDet state and initiates a new search. This cycle is performed until the object is found, or the user aborts the action within the UCB state. If no object is found in the ObjDet state, the FSM switches to the *Visual Attention* (VisAtt) state, that is also used for the localization of unknown objects.

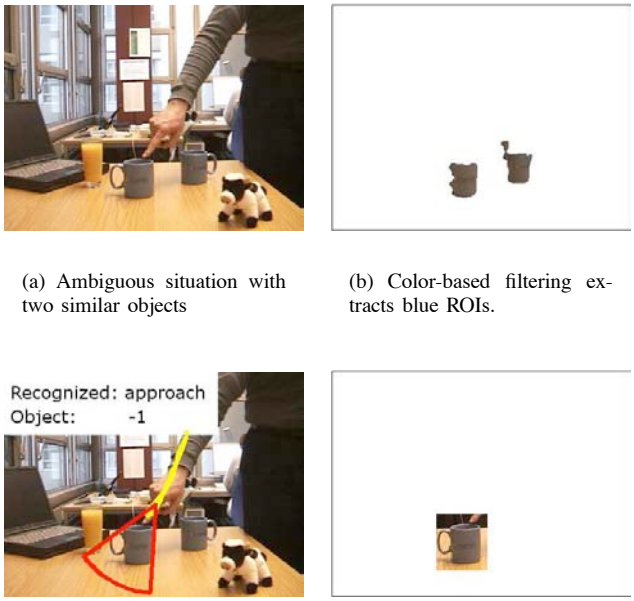
B. Unknown objects

If no object detection is available the OAS uses different filters in the following called *attention maps* that are similar to the attention maps described in [19]. The use of these filters is coordinated within the VisAtt state. They bring out salient image features like distinctive colors. The appropriate attention map is selected based on the additional verbal information (e.g., the color “blue”) given by the user. If the referenced object has several colors, we use the GrabCut algorithm [20] to extract a segmented view of the object. Since the gesture recognition also provides the direction of a pointing gesture, a bounding box in front of the hand position is set around the image area that contains the color which is supported by the attention map. Finally, a view of the blue cup is extracted from the scene.

If the verbal information given by the user is insufficient to determine a ROI, the FSM changes to the UCB state. In the UCB state the OAS sends a request to the dialog module in order to get more information about the object which the user refers to. The UCB state is also reached if more than one ROI is found. Then, the OAS asks the dialog component to resolve this ambiguity. When the dialog component has sent a response to the OAS, the FSM returns to the VisAtt state. As soon as the OAS has determined the ROI, the FSM switches to the ObjAna state to acquire the position of the object based on the hand position of the user. Next, the FSM switches to the ObjStor state, to store the extracted view and the position of the object in the scene model. Then, the FSM returns to the object alertness state to await new orders.

VI. Results

In our experiment BIRON was placed in front of a table with some objects on it (see Fig. 6(a)). As one example the robot was instructed with the phrase “This is Britta’s blue cup”. Referring to the image there are two blue cups in the scene, so an ambiguity exists. In order to extract all possible ROIs, an attention map that highlights the color ‘blue’ corresponding to the verbal input is selected. Consequently, two possible ROIs are extracted in the color-filtered image as shown in Fig. 6(b). This ambiguity is resolved by evaluating the user’s gesture as depicted in Fig. 6(c). The line above the users arm in the image denotes the hand’s trajectory and the circle segment marks the search area depending



(a) Ambiguous situation with two similar objects

(b) Color-based filtering extracts blue ROIs.

(c) Visualization of gesture information.

(d) Object-view obtained by using ROI and gesture data.

Fig. 6. Resolving ambiguities by color and gesture evaluation.

on the pointing direction. As the cup is not a previously known object, the pointing gesture of the hand is detected without an object in its search area. But, based on this 'approach' movement of the user, an object is expected in the movement direction of the hand. This enables the system to set a bounding box surrounding the blue-colored region in the search area, resulting in a view of the blue cup to which the user refers to (see Fig. 6(d)). In this way a new object view is learned and can be used for the template-based NCC object recognition algorithm from now on.

VII. Summary

In this paper we presented a detailed description of the multi-modal object attention system used on our robot companion BIRON. Through incorporating gestures and verbally specified object properties, the robot can detect known and unknown objects. Newly acquired object information is stored in a scene model for later retrieval. This growing knowledge base improves the interaction quality as the robot can more easily focus its attention on objects it has been taught previously. The ability to learn about an unknown environment allows the robot to be used as a companion in private homes.

References

[1] F. Lömker and G. Sagerer, "A multimodal system for object learning," in *Pattern Recognition, 24th DAGM Symposium, Zurich, Switzerland*, ser. Lecture Notes in Computer Science 2449, L. V. Gool, Ed. Berlin: Springer, 2002, pp. 490–497.

[2] F. Kaplan and V. V. Hafner, "The challenges of joint attention," in *Proc. 4th Int. Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic System*. Lund University Cognitive Studies 117: In Berthouze, L. and Kozima, H. and Prince, C. and Sandini, G. and Stojanov, G. and Metta, G. and Balkenius, C., 2004, pp. 67–74.

[3] C. Breazeal, *et al.*, *Humanoid Robots as Cooperative Partners for People*, *Int. Journal of Humanoid Robots*, 2004, in press.

[4] A. Utsumi, N. Tetsutani, and S. Igi, "View-based detection of 3-D interaction between hands and real objects," in *In Proc. Int. Conf. on Pattern Recognition*, vol. 4, Cambridge, UK, 2004, pp. 961–964.

[5] S. S. Ghidary, Y. Nakata, H. Saito, M. Hattori, and T. Takamori, *Multi-Modal Interaction of Human and Home Robot in the Context of Room Map Generation*, *Autonomous Robots*, vol. 13, no. 2, pp. 169–184, 2002.

[6] A. Haasch, *et al.*, "BIRON – The Bielefeld Robot Companion," in *Proc. Int. Workshop on Advances in Service Robotics*, E. Prassler, G. Lawitzky, P. Fiorini, and M. Hägele, Eds. Stuttgart, Germany: Fraunhofer IRB Verlag, 2004, pp. 27–32.

[7] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer, "Supporting Advanced Interaction Capabilities on a Mobile Robot with a Flexible Control System," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, Sendai, Japan, 2004, pp. 3649–3655.

[8] J. Fritsch, M. Kleinehagenbrock, A. Haasch, S. Wrede, and G. Sagerer, "A flexible infrastructure for the development of a robot companion with extensible HRI-capabilities," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, Barcelona, Spain, 2005, pp. 3419–3425.

[9] S. Lang, *et al.*, "Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot," in *Proc. Int. Conf. on Multimodal Interfaces*, Vancouver, Canada, 2003, pp. 28–35.

[10] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Int. Conf. on Advanced Robotics*, Coimbra, Portugal, 2003, pp. 317–323.

[11] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories: CONDENSATION-based recognition of gestures and expressions," in *Proc. European Conf. on Computer Vision – Volume I*, ser. Lecture Notes in Computer Science. Freiburg, Germany: Springer-Verlag, 1998, vol. 1406, pp. 909–924.

[12] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *ICCV'98*, Mumbai, India, 1998, pp. 107–112.

[13] N. Hofemann, J. Fritsch, and G. Sagerer, "Recognition of deictic gestures with context," in *Pattern Recognition; 26th DAGM Symposium, Tübingen, Germany, Proc.*, ser. Lecture Notes in Computer Science, C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, and B. Schölkopf, Eds. Heidelberg, Germany: Springer-Verlag, 2004, vol. 3175, pp. 334–341.

[14] B. Möller, S. Posch, A. Haasch, J. Fritsch, and G. Sagerer, "Interactive object learning for robot companions using mosaic images," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005, to appear.

[15] S. Wrede, M. Hanheide, C. Bauckhage, and G. Sagerer, "An Active Memory as a Model for Information Fusion," in *Proc. Int. Conf. on Information Fusion*, vol. 1, Stockholm, Sweden, 2004, pp. 198–205.

[16] J. P. Lewis, "Fast template matching," in *Proc. Conf. on Vision Interface*, Quebec, Canada, 1995, pp. 120–123.

[17] L. Fei-Fei, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proc. Int. Conf. on Computer Vision*, vol. 2, Nice, France, 2003, pp. 1134–1141.

[18] D. R. Wilson and T. R. Martinez, *Reduction Techniques for Instance-Based Learning Algorithms*, *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.

[19] L. Itti, C. Koch, and E. Niebur, *A Model of Saliency-based Visual Attention for Rapid Scene Analysis*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[20] C. Rother, V. Kolmogorov, and A. Blake, *GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts*, *Proc. Siggraph*, 2004.

Interactive Object Learning for Robot Companions using Mosaic Images

B. Möller and S. Posch
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg
06099 Halle/Saale, Germany
moeller@informatik.uni-halle.de

A. Haasch*, J. Fritsch*, and G. Sagerer*
Faculty of Technology
Bielefeld University
33594 Bielefeld, Germany
ahaasch@TechFak.Uni-Bielefeld.DE

***Abstract*— Natural *human-robot interaction* (HRI) is a key feature of mobile robot companions collaborating with humans. To achieve natural HRI, multiple communication modalities like vision, speech, and gestures have to be utilized. Besides, capabilities to emulate cognitive processes, e.g., object learning and object recognition, are essential. In this work we present a new approach to interactive object learning enabling multi-view object representation. To overcome a robot's limitation of having only one view point, we make use of an iconic memory consisting of previously acquired images. As the relevant scene area is unknown during construction of the iconic memory, a representation in the form of *mosaic images* is applied. The relevant image patches describing an object referenced by the user are selected through an *object attention mechanism*. The resulting multi-view object representations improve the flexibility of our interactive approach for object learning.**

***Index Terms*— human-robot interaction, robot companion, object learning & recognition, iconic memory, mosaic images**

I. Introduction

Achieving a high degree of natural interaction between a robot and a user is an ambitious research goal. The basic idea of such a *robot companion* is that it 'lives' together with the user in his private home. Therefore, the user interface has to be designed as intuitive to use as possible. To solve this task, the robot must be able to react to the user's actions, like deictic gestures or verbal utterances, in order to enable a knowledge transfer similar to human-human interaction.

As robot companions are intended primarily for private homes, they must be able to learn new environments. This task is reflected in the so-called *home tour scenario*. The idea of this scenario is that after the user has bought a robot in a store, he takes it to his home and shows the robot all locations and objects that are important for later use. Thus, the robot has not only to be able to detect potential communication partners, but also to recognize objects and locations that are important during the interaction. In previous work, we presented a mechanism for controlling a robot's attention to detect communication partners [1]. With this ability to start an interaction with a user, learning of new objects and recognizing objects previously learned become very important capabilities for a robot companion to act in a private home.

In this paper, we focus on the acquisition of object information during an interactive learning phase. We present an *object attention system* (OAS) that enables the robot to focus its attention on objects that are referred to by the user. Subsequently, the system extracts their relevant properties for recognizing them autonomously from now on. In order to focus the attention of the robot on objects, it is necessary to distinguish between known and unknown objects. This is necessary as it cannot be assumed that every possible object is known by the robot a priori. Especially learning unknown objects is a difficult task because the corresponding methods are usually view-based and, therefore, depend on visual object features.

To make the appearance-based recognition of learned objects more flexible it is useful to incorporate different views of an object. Therefore, a mobile robot would need to maneuver around the object to acquire different views of it. However, before being attended to an object the mobile robot is not always busy and can explore its environment collecting visual information. We present here a solution to accomplish this based on *mosaic images* that support efficient representations of image sequences acquired with an active camera. The capturing of mosaic images at different positions of a room enables the robot to gain several distinctive views of all the objects present. One great advantage of this procedure is that it does not interfere with ongoing interactions. Capturing different localized views during an interaction would lead to very time-consuming actions on the part of the robot and severly restrict the robot's interactivity. Additionally, it cannot be ensured that every single captured image contains a complete object and therefore merging several images might become necessary. Using the mosaic representation as an *iconic memory*, the associated image patches from other viewing directions can be extracted later on to obtain a

*This work has been supported by the European Union within the 'Cognitive Robot Companion' (COGNIRON) project (FP6-IST-002020) and by the German Research Foundation within the Collaborative Research Center 'Situated Artificial Communicators' as well as the Graduate Program 'Task Oriented Communication'.

In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2005. IEEE.

© IEEE. Personal use of this material is permitted.However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: IEEE Intellectual Property Rights Office / IEEE Service Center / 445 Hoes Lane / Piscataway, NJ 08855-1331 / phone: (732) 562-3966 / fax: (732) 981-8062

multi-view object representation. This method improves the recognition of learned objects as a larger number of different views is available to train object recognition algorithms.

The remainder of this paper is organized as follows: In the next section we will summarize earlier work related to our approach. Section III describes hardware details of our mobile robot platform BIRON and Section IV outlines the object attention system. Basics of mosaic image retrieval are discussed in Section V and Section VI gives details of how to access mosaic data for object recognition purposes. Exemplary results are presented in Section VII before the paper finishes with a conclusion.

II. Related Work

The interaction between a human and a robot through gestures and speech as well as learning of the communicated information has been investigated in a variety of contexts.

For example, the robot Leonardo [2] learns what to do with pushbuttons which are arranged around him. It can learn labels for each button by having a person point to a specific button and name it. Then, Leonardo can push those buttons that a person wants him to push. To detect all buttons in the robots vicinity, the system needs a camera which views the scene from above. In a similar setup with multiple cameras which have a view on the scene from above, Utsumi et al. [3] analyze interactions between human hands and objects. Here, no prior information about objects is needed, as objects are dynamically modeled based on their appearance.

While the approaches mentioned are characterized by a static setup and cameras are mounted on the ceiling, a robot companion is moving around in the environment and senses the scene only with its on-board camera. In [4] the Perseus architecture is used to interpret pointing gestures from a human to locate objects and store them in a long term visual memory. The system requires a whole person to be in its field of view to detect a gesture. Moreover, objects must contrast with the background in order to be extracted properly from the scene. Objects are not visually identified but only detected. Ghidary [5] presents a mobile robot that learns objects through the detection of the posture of a human's hand. Object views are stored in a topological map, but the views are always squares and, more importantly, only one view is stored in each training phase.

To overcome the limitation of acquiring only one view of an object, an iconic memory containing other scene views can be used to extract multiple views. Instead of using a camera with a wide field of view resulting in coarse image patches, the use of an active pan-tilt camera and the integration of the image sequence into one single image frame called *mosaic image* is advantageous. Mosaic images originate from the telecommunications since they allow for efficient compression of iconic data, e.g., video sequences [6]. Within the research field of robotics mosaic images have thus far mainly been used to solve localization and navigation tasks (e.g., [7]). Here, they are usually adopted to generate an iconic reference representation offline in order to compare it to acquired views later on. Contrary, adopting mosaic images as iconic memories with interactive systems requires mosaicing in an online fashion, i.e., integrating new image data into an evolving mosaic immediately. This is mainly due to limited resources of mobile systems and because sudden task switches might be necessary due to user interaction. Hence, widely used approaches where complete sequences are registered in an offline fashion and are then explicitly stored for later view rendering (e.g., [8], [9]) cannot be transferred to the area of application within this work.

III. Robotic Platform

Implementation and development of the approach outlined in this paper is performed using our mobile robot platform BIRON (Fig. 1), a Pioneer PeopleBot from ActivMedia™. The robot is equipped with an on-board PC (Pentium III, 850 MHz) for controlling the motors, on-board sensors, and for sound processing. An additional PC (Pentium III, 500 MHz) inside the robot is used for image processing. Both PCs, each running Linux, are linked by 100 Mbit Ethernet to a wireless LAN router. An additional laptop (Pentium M, 1.4 GHz) is linked via wireless LAN to perform additional computations and to enable remote control of the mobile robot.

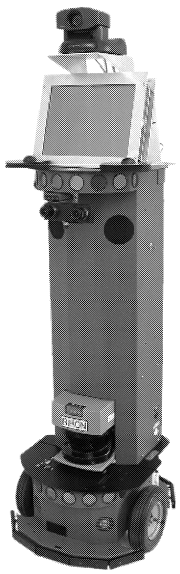


Fig. 1. BIRON.

Locating communication partners and analyzing a scene is accomplished adopting different modalities and hence varying sensor devices. A pan-tilt color camera (Sony EVI-D31) is mounted on top of the robot at a height of 141 cm for acquiring images of the upper body part of humans interacting with the robot. It is also used for the construction of mosaic images. Two AKG far-field microphones are located at the front of the upper platform at a height of 106 cm, right below a touch screen display. They enable speech processing and especially stereo-based speaker localization. Below the upper sonar ring, a wide-angle VIDERE stereo camera is mounted at a height of 95 cm for visual localization. Finally, distances within the scene and to humans are measured facilitating a SICK laser range finder mounted at the front at a height of 30 cm.

The control of all components on BIRON and their communication is realized using the SIRCLE framework [10] resulting in a robot companion with basic interaction capabilities [11]. In order to enable BIRON to learn and to recognize objects, we recently developed an object attention system. It is activated on demand when the user is referring to an object

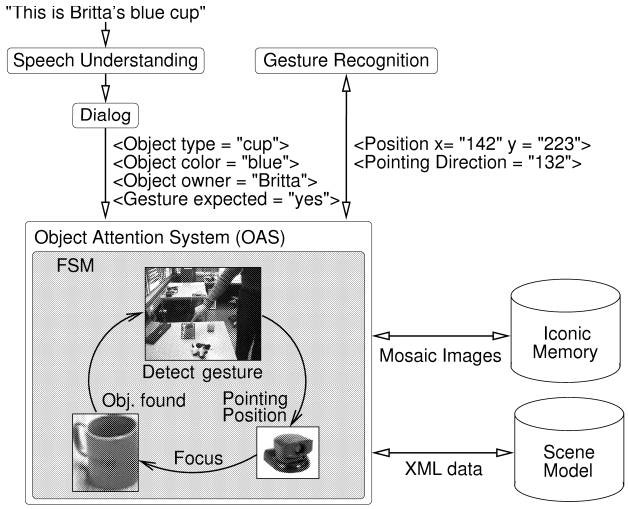


Fig. 2. Processing example of a multi-modal object reference by the object attention system.

or the robot has to interact with an object autonomously. In the following some fundamental aspects of this attention system will be outlined before the acquisition and use of mosaic images is explained in subsequent sections.

IV. Object Attention System

Opposed to many other approaches to realize a visual attention system, a mobile robot having a camera with a limited field of view needs some kind of sensor control to reorient the camera to a relevant part of the current scene. In the context of the home tour scenario outlined above the object to which a user refers to is considered important. However, private homes contain a tremendous variety of different objects. Determining the object the user refers to requires to recognize known objects and, more importantly, to interactively learn unknown objects. In order to accomplish this task, the object attention system (OAS) processes input from the camera, the dialog, and the gesture recognition module which is briefly described in the following. A more detailed description is presented in [12]. The coordination of the input modalities and the control of the hardware components (e.g., the pan-tilt camera) is realized by a finite state machine (FSM). This FSM controls the processing of the OAS (see Fig. 2).

In order to represent the objects in the environment of the robot companion, a knowledge base called *scene model* is used. This scene model is an active memory [13] and can be considered as a mixture of short-term spatial memory and long-term object memory. It stores not only the current position of an object, but also the object's properties that are extracted from the scene (e.g., its visual appearance) and given verbally by the user (e.g., the owner of the object).

The processing of the OAS is started when the dialog sends a request to identify an object in the scene that has been referenced by the user. As a simple example, a lexical cue like "this" or "that" triggers the speech understanding component to determine that a gesture is expected. As a consequence, the dialog provides the OAS with corresponding information and the gesture recognition module [14] is activated by the FSM. After finding a pointing gesture, the camera is reoriented based on the hand coordinates and the pointing direction that are provided by the gesture recognition module. This information is also used to restrict the region of interest (ROI) in the camera image that needs to be searched for the object referenced by the user. If the dialog module sends the description of the object (e.g., type, color, owner, see Fig. 2), an inquiry to the scene model is initiated. The object is considered as known if the scene model already contains an object with this description, otherwise the object is unknown. The two related processing strategies for localizing known and learning unknown objects are described in the following two sections using the example "This is Britta's blue cup".

A. Recognizing known objects

The search for a known object type involves an object detection process. For this task we use an object recognizer based on the fast Normalized Cross-Correlation (NCC) algorithm described in [15]. Several other approaches that could be used to recognize objects can be integrated as well (e.g., [16], [17]). However, as we focus on the interactive object learning and not on the issue of object recognition, the simple NCC is sufficient here.

In the case when the scene model contains a known object with properties matching the verbally referenced object (e.g., color=blue, type=cup, owner=Britta), the OAS fetches appropriate image patterns for recognizing the cup from the scene model and the object detection is initiated. If the object is found within the ROI, a success message is sent to the dialog module and the obtained position of the object is stored in the scene model. If two or more objects of the same type are detected in the camera image, a message is send to the dialog component to find out which of the objects is referenced by the user.

B. Finding unknown objects

For detecting and learning unknown objects we assume that an accompanying pointing gesture is present and a query to the scene model is not successful, i.e., there is no object with matching properties stored in the scene model. In this case, the OAS uses several different predefined color filters that are similar to the attention maps described in [18]. Based on the additional verbal information given by the user (e.g., "blue") an appropriate color filter is selected. Subsequently, a bounding box in front of the hand position is set around the image area that contains the color which is supported by the corresponding attention map. This bounding box is used to extract a view of the blue cup from the camera image. If the referenced object has several colors, we use the GrabCut algorithm [19] to extract a segmented view of the object.

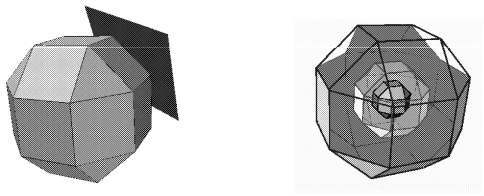


Fig. 3. Polytopial coordinate frame (rhombicuboctahedron) with focus image plane attached (left) and sketch of the hierarchical structure of the visual scene memory (right).

However, this single object view is not sufficient to recognize the cup from other viewing directions later on. Therefore, the following section presents a technique to extract high resolution mosaic images from a scene. The use of these iconic scene representations for extracting multiple views of an object is described in Section VI.

V. Mosaic Image Retrieval

Mosaic images yield an efficient approach to compactly represent image sequence data acquired with an active camera. The fundamental idea is based on the assumption of redundant information within an image sequence captured with a single camera. By exploiting these redundancies correspondences between different images of a sequence can be detected. This allows to align different images to each other *(registration)*. The correspondences are usually encoded by transformations allowing to warp the images towards a common coordinate frame *(reference frame)*. Subsequently, one single mosaic image can be generated by fusing the color information of individual pixels from all warped images *(integration)*. Since the mosaic represents all scene parts ever visible in any of the processed images it extends the camera's field of view in space and time.

When applying mosaic images as an iconic memory within interactive systems special algorithms and data structures are required. For example, a mosaic-based memory should not restrict the interactivity of the overall system. Hence, access to the memorized data has to be provided any time during processing the images. In addition, interactive and mobile systems usually do not provide enough capabilities to store a complete sequence and to process all images simultaneously. As a consequence and in contrast to several other mosaicing approaches (e.g., [8], [20], [9]) this implies that incoming data has to be processed in an *online* fashion. Each image has to be registered and integrated into the evolving mosaic image as soon as it becomes available.

Providing easy access to mosaic data in interactive systems not only requires online processing strategies, but also enforces to support the application of conventional image analysis techniques to the represented data. Since image analysis and also object recognition techniques are usually based on Euclidean coordinates this has to be considered in computation of the mosaic images and particularly in choosing an appropriate reference frame for projecting the images. Reference frames for mosaics are mainly determined by scene structure and the degrees of freedom of the camera. Representing visual data of indoor scenes can generally be done by storing panoramic views from different positions within a scene acquired with a *stationary*, but *rotating* and *zooming* camera. While single Euclidean planes yield large distortions when applied as reference frames for representing data from such cameras, spheres are a common choice. However, they render the application of existing image analysis algorithms impossible. Besides, representing mosaics in spherical coordinates without keeping all images (cf. [8], [9]) and online registration of new image data are also difficult tasks within this framework.

A. Polytopial Coordinate Frames

To cope with the problems outlined above, the reference frames of our mosaics consist of a set of individual tiles regularly arranged around the optical center of the camera tangentially to a sphere. Their global arrangement is derived from *polytopes* (Fig. 3). Each tile is equipped with a local 2D coordinate system supporting the direct application of conventional image analysis techniques. Image data is represented within this reference frame by projecting data onto the tile meeting the current camera orientation best and hence minimizing distortions.

Since a natural scene usually contains visual data of different levels of detail and hence of varying interest, most cameras provide zooming capabilities. Within the outlined home tour scenario the scene needs to be scanned in coarse resolution to get an overview. At the same time recognition of an object can only be done based on as much detail as possible implying to zoom in on the object. Due to these requirements an iconic representation based on mosaic images should not be restricted to a single resolution. This is realized by nesting differently scaled instances of a polytope into each other (Fig. 3). According to the focal length of the input image data the polytope instance is chosen for data projection that meets the resolution of the data best. The focal length results either from self-calibration algorithms [21] or can be provided by look-up tables generated in advance facilitating an offline calibration strategy [22]. According to the outlined multiple plane and multi-resolution structure the mosaic images are called *multi-mosaics* (cf. [23], [24]).

B. Online Registration and Integration

The motion of a stationary, but rotating and zooming camera can adequately be modeled using projective transformations, also called *homographies* (e.g., [22]). Their parameters are estimated using the projective flow approach [25] which basically includes optical flow computations between a pair of images restricted by the projective motion model. To achieve robust transformation parameters even in a long-term online computation, registration is done in frame-to-mosaic mode [26]. Each new image is registered to the evolving

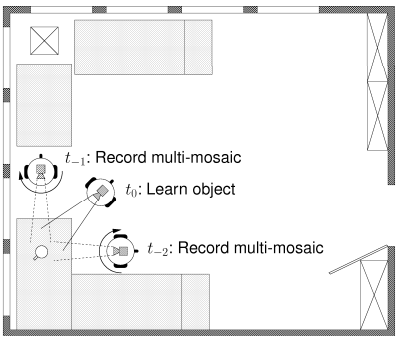


Fig. 4. Capturing different views of objects through mosaic images.

mosaic incorporating all images registered earlier. Image integration is accomplished by copying the data from new images directly to the mosaic. To avoid visible transitions (seams) at region boundaries images are blended along boundaries using linear or sigmoid blending functions.

During registration and integration of new images as well as when accessing and analyzing data represented in the multi-mosaic the geometric structure of the underlying polytopial reference coordinate frame has to be considered. Although multi-mosaics support an efficient online representation of iconic data due to their piecewise planarity further improvements are possible by simplifying the handling of discontinuities between different tiles. Therefore, we adopt an additional image plane, the *focus image plane* (FIP). It is attached to the polytope and hides its actual structure (Fig. 3). It also traces the movements of the camera and changes of focal length. New images are registered and integrated in reference to this plane and image data is only copied to the tiles themselves on update situations. These updates occur if new image data cannot be integrated completely into the FIP any longer, enforcing a change of its position and orientation. Then, iconic data represented on the FIP is projected onto corresponding tiles, the plane's position and orientation are updated according to the current setting of the camera, and finally new reference data is projected back onto the new FIP before mosaicing continues. Since the FIP is usually chosen two to three times larger in size than single images several images can be integrated before an update of the FIP becomes necessary. Hence, the FIP supports efficient online computation of mosaic images.

VI. Mosaic Data for Object Recognition

When an object has been focused by the OAS, interactive recognition and learning procedures are invoked which adopt appearance-based approaches. The more iconic data is available, the more flexible these algorithms work. An example demonstrating the use of mosaic images to represent iconic scene data and hence providing different object views for recognition and learning purposes is shown in Fig. 4. While BIRON was waiting for a communication partner it captured two multi-mosaics at different positions within a scene, in Fig. 4 indexed with timestamps t_{-2} and t_{-1}. As soon as its attention is addressed by a human and guided to a specific object (at time t_0) object attention mechanisms are invoked as described in Section IV to recognize or learn the intended object based on its appearance. All mosaic images formerly acquired are now available to be checked for additional views of this object (cf. Fig. 4). For this purpose, the object's 3D position has to be derived from the intended ROI as well as BIRON's current position and viewing direction.

A coarse approximation of the object's shape is sufficient for extracting suitable views from the different multi-mosaics and hence objects are approximated by cuboids. The distance of the object referenced is estimated using depth data from the stereo camera averaged over a small area at the center of the object. Now, a cuboid volume occupied by the object referenced can be calculated by simple 3D geometry.

Given this 3D volume and the 3D positions of all multi-mosaics in a common coordinate system a view of the object can be extracted from each mosaic providing corresponding data. For a single multi-mosaic this is accomplished by projecting the eight corners of the object's 3D volume onto the tiles of this mosaic. Calculating the bounding rectangles of the resulting projections on each tile yields the relevant visual data contained in the multi-mosaic. From these data conventional images are generated by collecting the data from related tiles. The resulting set of images extracted from all mosaics is subsequently passed to the OAS. Depending on the type of object recognizer applied, these images can be used to train a new object model. In our application the images are directly stored in the scene model for the appearance-based NCC recognition algorithm.

VII. Results

We demonstrate our approach using the prototypical example of section IV. The multi-mosaic captured at position t_{-1} in the office sketched in Fig. 4 is shown in Fig. 5. It is based on a sequence consisting of 30 images covering a camera rotation of approximately 60^o. Due to a sparse memory representation only polytope tiles that actually contain data are represented internally and are thus depicted. The unknown object was referenced with a sample utterance "This is Britta's blue cup" and a gesture. As described in section VI the approximated 3D volume of the object is projected onto the multi-mosaics resulting in the red square shown in Fig. 5. This subimage is shown in Fig. 6(a) and the corresponding subimage from the second mosaic in Fig. 6(c). To extract more accurate views of the object an attention map that highlights the color 'blue' proper to the verbally specified color is selected. These images are presented in Fig. 6(b) and 6(d). They are immediately available to build an appearance-based model for the formerly unknown object. Depending on the idle time before, a larger number of mosaics and thus more views of the object can be supplied.

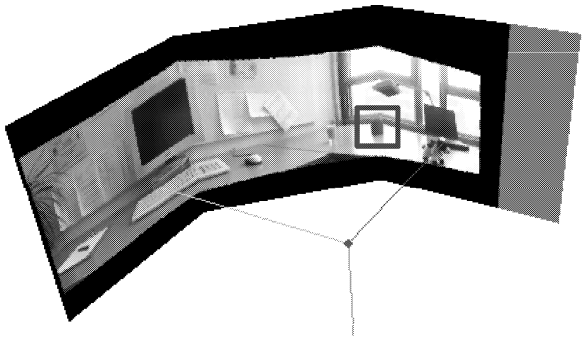
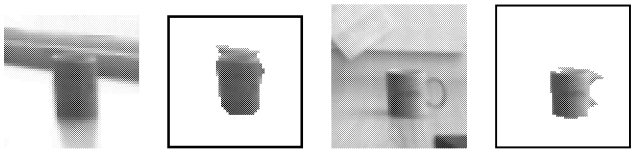


Fig. 5. Example multi-mosaic generated from a 60^o camera pan.



(a) (b) (c) (d)

Fig. 6. (a), (c) Subimages extracted from the mosaics; (b), (d) Patches after applying the attention map for color blue.

VIII. Summary

In this paper we presented an approach to improve the cognitive capabilities of our robot BIRON with regard to object learning. For this purpose we apply an iconic scene memory based on mosaic images. It allows to store visual data acquired during scene exploration for later use with object recognition tasks. Especially, appearance-based object recognition approaches widely used benefit from such a memory. Usually their performance is directly correlated with the number of views available. We achieve a large number of training images by extracting views from different mosaics located at different positions within a scene. Consequently, the enhancement of the object attention system with an iconic memory based on mosaic images enables larger flexibility in object recognition tasks improving the overall performance of a robot companion in human-robot interaction.

References

[1] S. Lang, *et al.*, "Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot," in *Proc. Int. Conf. on Multimodal Interfaces*, 2003.

[2] C. Breazeal, *et al.*, *Humanoid Robots as Cooperative Partners for People*, *Int. Journal of Humanoid Robots*, 2004.

[3] A. Utsumi, N. Tetsutani, and S. Igi, "View-based detection of 3-D interaction between hands and real objects," in *In Proc. Int. Conf. on Pattern Recognition*, vol. 4, Cambridge, UK, 2004, pp. 961–964.

[4] R. E. Kahn, M. J. Swain, P. N. Prokopowicz, and R. J. Firby, "Gesture recognition using the perseus architecture," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Francisco, CA: IEEE Computer Society, 1996, pp. 734–741.

[5] S. S. Ghidary, Y. Nakata, H. Saito, M. Hattori, and T. Takamori, *Multi-Modal Interaction of Human and Home Robot in the Context of Room Map Generation*, *Autonomous Robots*, vol. 13, no. 2, pp. 169–184, 2002.

[6] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, *Efficient representations of video sequences and their applications*, *Signal Processing: Image Communication*, vol. 8, no. 4, pp. 327–351, 1996.

[7] H. Ishiguro, T. Maede, T. Miyashita, and S. Tsuji, "A strategy for acquiring an environmental model with panoramic sensing by a mobile robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 724–729.

[8] G. Bishop and L. McMillan, "Plenoptic modeling: An image-based rendering system," in *Proc. Int. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Los Angeles, 1995, pp. 39–46.

[9] H.-Y. Shum and R. Szeliski, *Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment*, *Int. Journal of Computer Vision*, vol. 36, no. 2, pp. 101–130, 2000.

[10] J. Fritsch, M. Kleinehagenbrock, A. Haasch, S. Wrede, and G. Sagerer, "A flexible infrastructure for the development of a robot companion with extensible HRI-capabilities," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, Barcelona, Spain, 2005, pp. 3419–3425.

[11] A. Haasch, *et al.*, "BIRON – The Bielefeld Robot Companion," in *Proc. Int. Workshop on Advances in Service Robotics*, E. Prassler, G. Lawitzky, P. Fiorini, and M. Hägele, Eds. Stuttgart, Germany: Fraunhofer IRB Verlag, 2004, pp. 27–32.

[12] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer, "A multi-modal object attention system for a mobile robot," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005, to appear.

[13] S. Wrede, M. Hanheide, C. Bauckhage, and G. Sagerer, "An Active Memory as a Model for Information Fusion," in *Proc. Int. Conf. on Information Fusion*, vol. 1, Stockholm, Sweden, 2004, pp. 198–205.

[14] N. Hofemann, J. Fritsch, and G. Sagerer, "Recognition of deictic gestures with context," ser. Lecture Notes in Computer Science, C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, and B. Schölkopf, Eds., vol. 3175. Heidelberg, Germany: Springer-Verlag, 2004, pp. 334–341.

[15] J. P. Lewis, "Fast template matching," in *Proc. Conf. on Vision Interface*, Quebec, Canada, 1995, pp. 120–123.

[16] L. Fei-Fei, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proc. Int. Conf. on Computer Vision*, vol. 2, Nice, France, 2003, pp. 1134–1141.

[17] D. R. Wilson and T. R. Martinez, *Reduction Techniques for Instance-Based Learning Algorithms*, *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.

[18] L. Itti, C. Koch, and E. Niebur, *A Model of Saliency-based Visual Attention for Rapid Scene Analysis*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[19] C. Rother, V. Kolmogorov, and A. Blake, *GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts*, *Proc. ACM Siggraph*, pp. 309–314, 2004.

[20] H. S. Sawhney, S. Hsu, and R. Kumar, "Robust video mosaicing through topology inference and local to global alignment," in *Proc. European Conf. on Computer Vision*, 1998, pp. 103–119.

[21] L. de Agapito, R. I. Hartley, and E. Hayman, "Linear self-calibration of a rotating and zooming camera," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, Fort Collins, CO, 1999, pp. 1015–1021.

[22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[23] B. Möller and S. Posch, "A mosaic-based visual memory with applications to active scene exploration," in *Proc. of Mirage*, INRIA Rocquencourt, France, 2005, pp. 117–125.

[24] B. Möller, D. Williams, and S. Posch, *Towards a Mosaic-based Visual Representation of Large Scenes*, *Int. Journal on Pattern Recognition and Image Analysis, Special issue*, vol. 14, no. 2, pp. 262–266, 2004.

[25] S. Mann and R. W. Picard, "Video orbits of the projective group: A new perspective on image mosaicing," MIT Media Laboratory Perceptual Computing Section, Tech. Rep. 338, 1996.

[26] P. Burt and P. Anandan, "Image stabilization by registration to a reference mosaic," in *Proc. ARPA Image Understanding Workshop*, vol. 1, Monterey, CA, 1994, pp. 425–434.

Human-style interaction with a robot for cooperative learning of scene objects*

Shuyin Li, Axel Haasch, Britta Wrede, Jannik Fritsch, Gerhard Sagerer
Faculty of Technology
Bielefeld University
33594 Bielefeld, Germany
{shuyinli, ahaasch, bwrede, jannik, sagerer}@techfak.uni-bielefeld.de

ABSTRACT

In research on human-robot interaction the interest is currently shifting from uni-modal dialog systems to multi-modal interaction schemes. We present a system for human-style interaction with a robot that is integrated on our mobile robot BIRON. To model the dialog we adopt an extended grounding concept with a mechanism to handle multi-modal in- and output where object references are resolved by the interaction with an object attention system (OAS). The OAS integrates multiple input from, e.g., the object and gesture recognition systems and provides the information for a common representation. This representation can be accessed by both modules and combines symbolic verbal attributes with sensor-based features. We argue that such a representation is necessary to achieve a robust and efficient information processing.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Intefaces—*Natural language*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Tracking, Object recognition*; H.2.5 [**Heterogeneous Databases**]: Heterogeneous Databases

General Terms

Management

1. INTRODUCTION

Multi-modality is one of the most important features that characterize human-human social interaction. Based on this observation, designers of computer systems have been trying to integrate multi-modal input and output mechanisms in human-machine interactions to enable more intuitive operations for human users. Given the huge amount of existing multi-modal systems with quite different notions of multi-modality we first want to discuss two important aspects of modality intuitiveness to clarify our position. In general it is assumed that the degree of intuitiveness of a modality will determine the smoothness and efficiency of the interaction it facilitates. Thus, the question is what modalities are the most intuitive for users? We argue that the answer is of evolutionary nature and is application-dependent. For example, the mouse has become the major modality in human-computer interaction for many users. For these users the mouse is probably one of the most intuitive ways to operate a computer although it has little to do with the natural communication channels they use in human-human communication such as speech. In contrast, many other users prefer to write commands directly. We can imagine that future generations will find other modalities more intuitive than a mouse. The feeling of intuitiveness in computer operation is thus continuously changing. Even the same people may judge the intuitiveness of one modality differently when operating different computer systems. For example, people tend to anthropomorphize mobile robots when interacting with them. We argue that this is even more the case for application areas where a robot is supposed to assist and accompany humans in a social environment such as private households. Our envisioned robot is supposed to "live" in a private household. Our long term goal, therefore, is to endow it with social capabilities so that it can become some sort of "companion". This means that we do not envision it to serve humans in a master-slave manner as people tend to suppose. Rather, a robot companion should *cooperate* with humans to achieve certain goals. One basic function of such a Robot Companion is to be able to learn interactively about its environment. We therefore devised the *home-tour* scenario within our project where a human user is supposed to show his/her home to a newly purchased robot.

Based on our goal to design a robot that can be accepted by the user as a companion we argue that the interaction with such machines should be in a human style. We define the term *human style modalities* as *multi-modal communication channels that humans are biologically equipped for and (learn to) use from their birth.* Typical examples are speech and gestures. These modalities differ from other modalities like mouse and keyboard in that they are learned nat-

*This work has been partially supported by the European Union within the 'Cognitive Robot Companion' (COGNIRON) project (FP6-002020) and by the German Research Foundation within the Collaborative Research Center 'Situated Artificial Communicators' as well as the Graduate Program 'Task Oriented Communication'.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMI'05, October 4–6, 2005, Trento, Italy.
Copyright 2005 ACM 1-59593-028-0/05/0010 ...$5.00.

urally and without the use of artificial devices. In contrast, we define artificial modalities that are commonly used for human-computer interaction as *virtual modalities* because their effect is a virtual one which is only observable by humans via an artificial interface (e.g., the computer display). Thus, since people tend to anthropomorphize robots by expecting human-like abilities and attributes we conclude that robots should be endowed with human-style modalities for interaction.

A further aspect concerns the knowledge representation. Consider our home-tour scenario where the interaction will involve a high degree of deictic activity as the user will point to diverse objects in the environment. Thus, when the user points to his/her computer and explains "This is my computer", the robot should be able to recognize the user's gesture, find the computer and associate the symbolic name "computer" with a visual representation. This knowledge should be stored in a multi-modal way in order to be retrievable from different modules for further interactions. Psychological theories of knowledge representation in humans suggest that the symbolic name of an object is associated with its sensory features like its image or haptic characteristics (e.g. [3]). When activating the name of an object other features of the object are also activated. This indicates that the cognitive representation of objects in humans is multi-modal and therfore allows for multi-modal processing of information. In order for a robot to cooperate with a human, it should therefore be able to also process multimodal information to build a representation similar to that of its human communication partner to support a better mutual understanding. We therefore developed a multi-modal representation scheme.

To summarize our position shortly: The intuitiveness of modalities needed for operation of computers and machines has an evolutionary aspect and depends on the individual applications. In our Robot Companion domain we are interested in human-style multi-modality that should be considered for both, communication channels and the representation of knowledge. Its impact is of functional and technical nature.

In this paper we will first present the multi-modal processing strategies of the Dialog System (section 4) and the Object Attention System (section 5) followed by a detailed description of our multi-modal representation scheme in section 6. Results in the form of a dialog example will be given in section 7.

2. RELATED WORK

While there is an increasing interest in multi-modal interfaces there is only a very limited number of applications that use human-style modalities based on an integrated multimodal knowledge representation.

That multi-modal cues are beneficial for increasing the robustness in human-robot interaction has been shown for example in [13] where communication errors are detected by using not only speech recognition scores but also by including laser data to infer the presence or absence of communication partners and noise sources. Repair actions also involve the use of multiple modalities by either driving around to actively search for a communication partner or by offering buttons as alternative communication channels in noisy environments. More human-style interactions have been suggested in [1] by modeling a *naturalness support behavior*. This behavior includes verbal strategies by inserting filler phrases, as well as non-verbal reactions such as nodding or head turning as reactions to environmental noise. The authors report positive reactions by the users but extensive evaluations still remain to be done. The benefits of using multimodalities in a learning scenario have been demonstrated in several applications. For example, the robot Leonardo [2] can learn the names of buttons when a human communication partner points them out by verbal and deictic instructions. Leonardo also learns specific interactions with these buttons by demonstration. However, while the interactive capabilities of Leonardo are quite realistic the underlying representations are simple and no new objects can be learned. An impressive system running on a mobile robot is presented by Ghidary et al. [6]. The robot is able to learn objects by analyzing speech commands and hand postures of the user. The user gives verbal information about the object's size and can describe the spatial relations between objects, e.g., by phrases like 'left of my hand'. The rectangular views of the learned objects are stored in a map representing the robot's environment and can be used for later interactions. Although the interaction system is very limited and the resulting map is rather coarse, this system can be compared to our approach as it also builds up a long-term memory about objects in the environment. However, we focus on a more detailed representation of objects and their later recognition in order to support natural human-robot interaction going beyond simple navigational tasks.

In general there is a tendency to either focus on building a robust representation while neglecting interaction smoothness or vice versa. We argue that in order to build a robot that is able to be perceived as a companion it needs both, a more natural interaction based on an integrated multi-modal representation.

3. OVERALL SYSTEM

The scene acquisition system described in this paper is being implemented on our mobile robot BIRON. BIRON's hardware platform is a Pioneer PeopleBot from ActivMedia with a pan-tilt camera for face tracking and object and gesture recognition, stereo microphones for speaker localization and speech recognition, and a SICK laser range finder for locating legs of potential communication partners. The overall architecture [10] of BIRON is based on a hybrid control mechanism and has three layers: a reactive, an intermediate and a deliberative layer (see Fig. 1). Modules that are responsible for reactive feedback of the system are set on the reactive layer: the *Person Attention System* detects potential communication partners and the OAS detects objects that users refer to. Since these are purely data-driven processes they belong to the reactive layer. Modules responsible for higher-level processing that involve top-down, expectation-driven strategies such as a planner or the Dialog System, are located on the deliberative layer. The Scene Model, which contains a multi-modal representation of the objects that the system has observed and can be seen as an intermediator between the Dialog System and the OAS, is consequently located on the intermediate layer. The communication between modules is carried out via XCF (XML Enabled Communication Framework). The system is centrally controlled by the so-called *Execution Supervisor* on the intermediate layer. It coordinates the module operations and makes sure that neither the reactive layer mod-

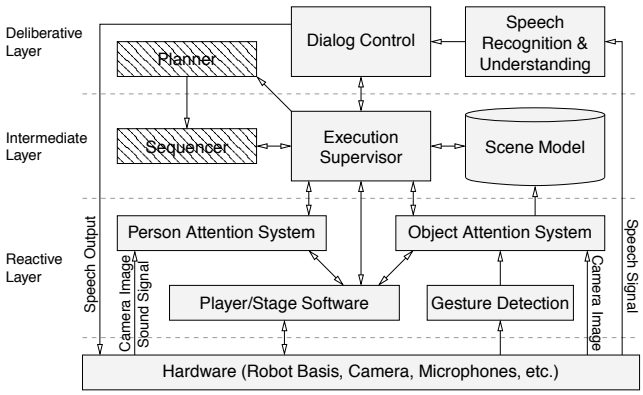


Figure 1: Overview of the BIRON architecture (optional modules currently not implemented are drawn in grey).

ules control the deliberative layer modules nor vice versa. Instead, it exerts control by taking into account the overall system state. This architecture allows for both fast reaction to dynamic environmental changes and extensive high-level planning and reasoning activities.

Since we focus on the interaction capabilities of BIRON, we do currently not integrate a planner and a sequencer. The Person Attention System establishes the basis for the interaction with users but is not involved in the process of resolving object references. In the following we therefore only describe the interfacing between the Dialog System, the OAS and the Scene Model.

4. THE DIALOG SYSTEM

The dialog system of BIRON is responsible for carrying out interactions with the user including handling miscommunications [16], guiding the discourse, and transferring user utterances to internal command for the robot control system to execute tasks. A dialog is made up of contributions from the dialog partners. Two central questions of dialog modeling are therefore (1) how to represent individual contributions represented and (2) how to represent the dynamic change of the dialog state represented which is triggered by individual contributions successively. In subsection 4.1 and 4.2 we are going to present our answers to these two questions. In section 4.3 we focus on the mechanism that the implemented dialog model provides for the integration of speech and visual input. A more detailed account on this integration will be given in section 5.

4.1 The structure of a contribution

Conversants contribute to a dialog in a multi-modal way. McNeill [12] investigated the relationship between speech and simultaneous conversational gesture and claims that the production of them are motivated by one single semantic source, the so-called "idea unit". Inspired by this finding, we represent the conversants' contribution as the so-called "interaction unit" that includes two important stages of the language production process. The structure of the interaction unit is illustrated in Fig. 2. An interaction unit has two layers: a *domain layer* and a *conversation layer*. The *domain layer* mirrors the cognitive activities of a dialog participant that motivate language production: If the interaction unit represents an utterance to be produced by the

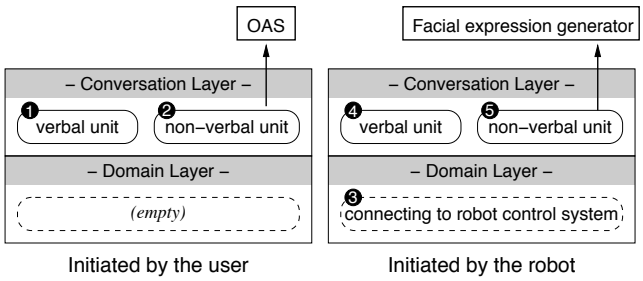


Figure 2: Processing flow in interaction units

robot itself the domain layer is where the Dialog System accesses the robot's control system or knowledge base. If the interaction unit represents an utterance of the user the domain layer remains empty because we do not make any assumptions about the user's cognitive activities behind the language front. In future work this may be replaced by a user model. The *conversation layer* transforms the intention that is created based on the results of these cognitive activities to language. For example, based on a successful follow behavior of the robot that is reported to the domain layer by the robot control system, the conversation layer formulates and synthesizes a message such as "OK, I follow you". The conversation layer consists of two units: a verbal and a non-verbal unit. Based on the intention that results from the domain layer they are responsible for generating output in verbal and non-verbal way, respectively.

The precondition of language production is successful language perception. Before reacting, i.e., before creating his/her own interaction unit to produce a contribution, a conversant first needs to understand the semantic meaning of his/her dialog partner's contribution by studying the verbal and non-verbal unit on the conversation layer of the dialog partner's interaction unit. Therefore the processing in the interaction unit should start from this language perception phase. Figure 2 illustrates how the robot processes user contributions. In our system, the user's verbal information as delivered by the Speech Understanding System initiates the creation of a user interaction unit (❶). In case that the user's intention can not be fully recognized by the verbal unit, the system will consult the visual perceptual module via OAS in the non-verbal unit (❷) and fuse these multi-modal information on the user conversation layer of the interaction unit. Once the user intention is fully recognized, the system creates an interaction unit for itself and tries to provide acceptance: the system first formulates and sends commands to the robot control system or the knowledge base on the domain layer (❸) and then generates verbal and non-verbal output on the conversation layer (❹, ❺) after receiving the execution results. Currently, we have implemented the visualization of facial expressions as the only non-verbal output. Thus, the integration of speech and visual information is mainly performed on the conversation layer of the interaction unit.

In the whole language perception and production process problems may occur, e.g., the semantic meaning of the user interaction unit cannot be resolved or the desired task cannot be executed by the robot system. These problems cannot be handled in a single interaction unit, new interaction units are necessary. Now the question arises as to how to organize the individual interaction units.

4.2 The grounding mechanism

The interaction units have to be organized in a dynamic way since every new contribution that is added to the dialog changes the dialog state. Our dialog model is inspired by the common ground theory of Clark [5]. According to this theory a dialog is carried out in the way that one participant presents an account (presentation) and the other issues the evidence of understanding of the account (acceptance). The grounding process is complete and both dialog participants can go on with a new account only if the acceptance is available. Dialog systems that implement this psychological model ([15], [4]) differ in their way of defining grounding units (the units of the discourse where the grounding takes place) and the organization of these units. We take exchanges in the style of adjacency pairs [14] as the grounding units. These exchanges consist of two interaction units that are initiated by the two dialog partners, respectively. The first interaction unit is the presentation and the second one is the acceptance, e.g., the first interaction unit represents a question and the second one the answer. To organize them we introduce four grounding relations between exchanges: (1) *default*: introducing a new task. A grounded default exchange has no further effect on the grounding of its preceding exchange. (2) *support*: clarifying in case of an ungroundable account. After a support exchange is grounded its initiator will try to ground the preceding one again that is updated with the new information. For example, clarification question in case of an incorrect speech recognition result. (3) *correct*: correct the previous account. As support, if such an exchange is grounded the initiator will try to ground the updated preceding one again. (4) *delete*: delete the previous account. If such an exchange is grounded, all the previous ungrounded exchanges can be deleted.

Each contribution is analyzed in terms of whether it is a presentation or an acceptance. If it is a presentation, then we also need to find out its grounding relation to the preceding exchanges. A presentation initiates the creation of an exchange that is put onto the top of a stack while an acceptance completes the top exchange of the stack. When the top exchange is complete it is popped. Additionally, actions like updating the preceding exchange can be triggered according to these relations. As long as there is an incomplete exchange on the top of the stack, the conversant other than the initiator of the exchange's presentation will try to ground it. The implemented dialog system enables us to handle clarification questions (as an exchange with support relation to its preceding exchange) and take initiative that is motivated by the robot control system. For example, in case of technical problems of the robot control system the implemented dialog system initiates an interaction unit to report this problem to the user. It does this by encoding the error message into its domain layer and generating output to the user on the conversation layer.

4.3 Resolving object references

In the following we detail how the Dialog System and the OAS cooperate to resolve object references in the user's utterances.

According to [9] there are three types of informational relations between gesture and speech: *reinforcement*, *disambiguation* and *adding information*. In our work, we focus on the "adding information" relation. When people use gestures to complete the meaning of their utterances they mostly indicate this intention in the utterance. For example, if a user says "This is my green mug" while pointing at a mug, the word "this" serves as a cue for the listener that he/she is using a gesture to specify the concrete location of the mug. But in case of the subsequent utterance "The mug is my favorite one" the listener usually does not expect a gesture but will search mentally in the dialog history which cup might be meant. According to these two different cases the Dialog System activates either the OAS or the Scene Model to resolve the object reference. This process can be illustrated as a UML activity diagram (see Fig. 3).

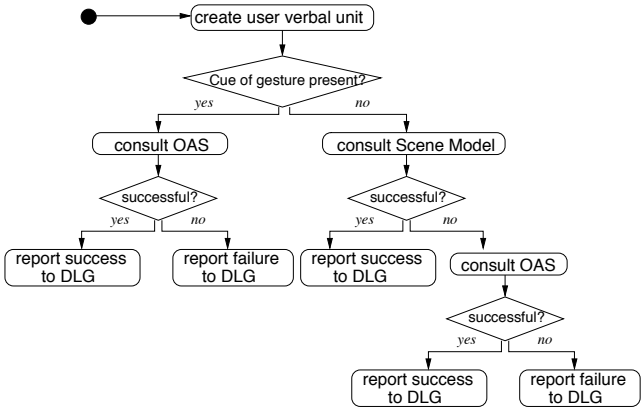


Figure 3: Resolving object references in user's verbal input (OAS: Object Attention System, DLG: Dialog System)

If there are any cues of the involvement of a gesture in the user's verbal input, e.g., the word "this" in the example above, this will be explicitely pointed out by the Speech Understanding System. The Dialog System interprets this hint as evidence that the user's verbal unit needs to consult the non-verbal unit. In the non-verbal unit the Dialog System activates the OAS by sending it the request to resolve the object reference "my green mug". The OAS activates the gesture recognition module. A successful gesture recognition result helps the OAS to orient the camera towards the position of the user's hand which enables the OAS to confine the Region Of Interest for its search of a green mug. This search is carried out, as the case may be, either by an appearance-based object recognizer or with the help of salient object features as described in section 5. Subsequently, the OAS sends the search result back to the Dialog System. In case of a positive result the OAS also updates the Scene Model with both symbolic and visual information about the new object "green mug". According to this result the Dialog System creates a system interaction unit (cf. Fig. 2) to provide acceptance for the user's input by either acknowledging or by initiating verbal repair.

If there is no evidence of the involvement of a gesture in the user's verbal input but only some objects to be identified (such as the "mug" in the example "The mug is my favorite one") the Dialog System will first try to find a corresponding entry in the Scene Model. The query is constructed with all the features of the object present in the current verbal input; in this case, the owner of the cup and his/her relation to this object (favorite). If the object can be found in the Scene Model the Dialog System finishes its processing on the conversation layer of the user interaction unit and creates an

acceptance to the user's input; if this object is not registered in the Scene Model, the Dialog System activates the OAS to find it in the current scene. This process is described in detail in the following section.

5. OBJECT ATTENTION SYSTEM

In order for a system to be able to acquire knowledge about objects in its environment it needs a mechanism to focus its attention on those objects that the user is currently talking about. In our context we define attention as the ability to select and concentrate on a specific stimulus out of all stimuli that are provided by the environment while suppressing others. The Object Attention System (OAS) therefore needs to coordinate the visual processing results (which currently consist of deictic gestures, object recognition results, and visual object features) and making them accessible for the Dialog System by storing them in the multi-modal Scene Model.

The OAS is activated when the user is verbally referring to an object and the Speech Understanding System has determined that either a gesture is expected or that the robot has to interact with an object autonomously. In order to acquire visual information about objects, BIRON uses an active camera with a maximal opening angle of view of about 50 degrees horizontal and 38 degrees vertical which facilitates only a limited field of view. It can therefore be necessary to re-orient the camera to relevant parts of the current scene which the user refers to. Once the robot has focused its attention on such a so-called *Region Of Interest*, the acquisition of information about this object, like position or view, can be completed. The Region Of Interest is that part of the image that has been specified by a gesture and contains the distinctive feature verbally specified by the user. Our assumption is that it contains an image of the object if the object is known to the robot. If it is unknown the Region Of Interest encloses the verbally specified visual feature (e.g. the "round thing" or a color).

The collected object data then has to be added to the robot's knowledge base which must allow retrieving stored object information and updating already stored data. Also, additional information given verbally by the user has to be stored. As this knowledge base, subsequently named as *Scene Model*, is crucial for the interaction between the Dialog System and the OAS because it represents BIRON's long-term memory, it is described in detail in section 6.

In addition to the maintenance of the Scene Model for memorizing tasks, the OAS needs to take care of the coordination of verbal information, gestures, and salient object features (e.g., color, shape, etc.) perceived by the camera, as well as the control of the hardware components like the camera during the object attention phases. This is realized by a *Finite State Machine* (see Fig. 5) where the input is mainly provided by the camera, the Dialog System (cf. section 4), the Speech Understanding System, and the gesture recognition component [7].

A crucial distinction that has to be made during the processing of multi-modal information is that between objects that are already known to the robot and those that are not (see Fig. 4). This is because in the latter case the OAS will have to establish (or 'learn') a first link between the verbal symbols describing the object and the percepts while in the former case the object needs to be retrieved from the database.

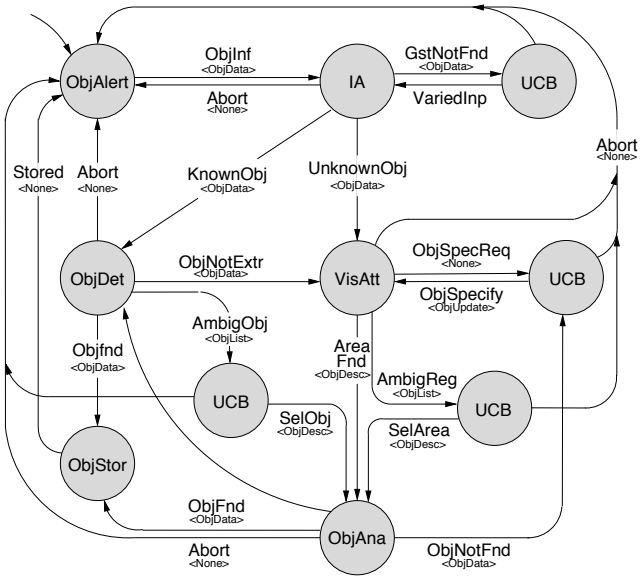


Figure 5: The Finite State Machine of the Object Attention System.

In both cases the OAS is activated on demand by the Dialog System if a gesture is expected or an access to the Scene Model by the Dialog System has failed afore. At this moment, the Finite State Machine (see Fig. 5) will be in the idle state *Object Alertness* (ObjAlert). Once the OAS is provided with data by the Dialog System, the Finite State Machine changes to the *Input Analysis* (IA) state. Now, the gesture recognition component is activated and provides the OAS with the user's hand coordinates and the direction of the corresponding pointing gesture. Thus, an area within the camera image is selected as the Region Of Interest. In case the Dialog System sends a description of the object (e.g., type, color, owner, etc.) to the OAS, a query to the Scene Model is initiated in order to check whether the object type is already known. In the following, we will describe in detail the processing for the case when the object type is known to the robot. The more complex process for the case of unknown objects will be exemplified subsequently.

5.1 Previously known objects

Suppose the user specifies an object type that the system has already stored in its Scene Model. In this case the Scene Model will return all object entries that match the symbolic description of the specified object. In order to verify if one of the returned objects is indeed the object the user refers to, the OAS will need to search for the object in the real scene and compare it with the stored image pattern. This search involves an object detection process for which we are currently using a simple appearance-based object recognizer that is only suitable for a very limited object scenario. It is based on the fast Normalized Cross-Correlation (NCC) algorithm described in [11] which is a simple but fast algorithm that is sufficient for our task at hand. However, in order for the system to work reliably in a more unstructured environment, as for example a real *home-tour* scenario a more sophisticated object recognizer will be needed.

After all appropriate image patterns have been retrieved for the known object type, the Finite State Machine switches

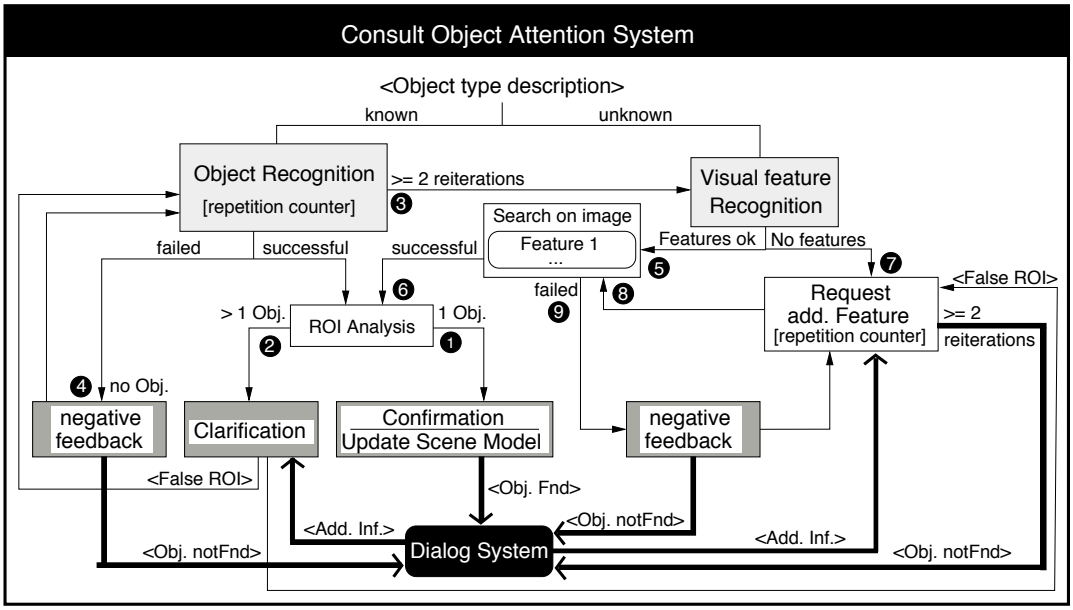


Figure 4: Schematic description of distinction between known and unknown objects

from the Input Analysis state to the *Object Detection* (ObjDet) state. Within the Object Detection state the OAS uses the retrieved image patterns (e.g., for cups) in order to feed them to the object recognizer. At the same time the camera is re-oriented based on the hand coordinates and the pointing direction that are provided by the gesture recognition component. Also, the position of the hand is used to determine the relevant Region Of Interest. Based on this information, the object detection process is initiated by scanning the Region Of Interest for patterns similar to those provided by the database. If an object is found by this procedure a confirmation message is sent to the Dialog System (cf. Fig. 4 (❶)) and the Finite State Machine switches to the *Object Store* (ObjStor) state. In this state the position of the object in the scene is updated in the Scene Model. Finally, the Finite State Machine returns to the Object Alertness state and the OAS awaits new orders.

If two or more objects of the same type are found in the real scene during the detection phase (cf. Fig. 4 (❷)) in the Object Detection state, the Finite State Machine will switch to the *User callback* (UCB) state. This means, that a message is sent to the Dialog System to clarify which of the found objects was meant by the user. After the Dialog System has provided a more detailed description, the Finite State Machine switches to the *Object Analysis* (ObjAna) state. In this state a new Region Of Interest is determined based on the information from the gesture detection and the extended verbal information. Now the Finite State Machine returns to the Object Detection state and initiates a new search. This cycle is performed until the object is found, or the user aborts the action within the User callback state but at most two times. Then, the Finite State Machine switches to the *Visual Attention* (VisAtt) state (cf. Fig. 4 (❸)). If no object is found in the Object Detection state (cf. Fig. 4 (❹)) this means that the user is referring to an unknown object which is supposedly similar to the description of objects previously retrieved from the Scene Model. However, since the object is unknown the Finite State Machine switches to the Visual Attention state, that is also used for the localization of unknown objects if two reiterations have been reached (cf. Fig. 4 (❸)).

5.2 Unknown objects

If no object detection is possible because no object entry matching the user's specification has been found in the Scene Model the OAS will search for salient features in the camera image such as colors or shapes by applying different filters that detect salient visual object features as specified by the user. We call these filters *attention maps* following the terminology of [8] where a similar technique is used. The use of these attention maps is coordinated within the Visual Attention state and can help to select Regions Of Interest. The appropriate attention map is selected based on the verbal information (e.g., the color) given by the user (cf. Fig. 4 (❺)).

Once a region matching the search criteria (i.e., color) is found within the Region of Interest by the attention map it is selected within a bounding box (cf. Fig. 4 (❻)). This bounding box is supposed to contain a view of the retrieved object (e.g., a blue cup) and is stored in the Scene Model (cf. Fig. 4 (❶)). Additionally, a confirmation message is sent to the Dialog System.

If the verbal information given by the user is insufficient to determine a Region Of Interest, that is if no visual descriptions are given that can be found by the attention map, (cf. Fig. 4 (❼)), the Finite State Machine changes to the User callback state. In the User callback state the OAS sends a request to the Dialog System in order to get more information (e.g., shape, position, ...) about the object which the user refers to. When the user has given a more specific description which is sent by the Dialog System to the OAS, the Finite State Machine returns to the Visual Attention state (cf. Fig. 4 (❽)). The User callback state is also reached if more than one Region Of Interest is found (cf. Fig. 4 (❷)). Then, the OAS asks the Dialog System to re-

solve this ambiguity. As soon as the OAS has determined the Region Of Interest, the Finite State Machine switches to the Object Analysis state to acquire the position of the object by means of the hand position of the user. Next, the Finite State Machine switches to the Object Store state and stores the extracted view and the position of the object in the Scene Model (cf. Fig. 4 (❶)). Then, the Finite State Machine returns to the Object Alertness state to await new orders.

If no Region of Interest is found during the search for visual object features, the Finite State Machine switches to the User Callback state and returns a negative response to the Dialog System. In parallel, the OAS asks the Dialog System for a more detailed object description and re-initiates a second search on the image (cf. Fig. 4 (❾)). If for a second time no Region Of Interest is found, the OAS sends a message to the Dialog System, that the search for the referenced object was not successful and returns into its idle state to await new orders from the Dialog System.

6. REPRESENTATION

Information acquired by the Dialog System and the OAS in the ongoing interaction with a user must be stored in an appropriate way. Because the same information from different modalities require different ways of representation the management of such a multi-modal database is a non-trivial task. Our approach to such a database, that we call *Scene Model*, is based on the concept of an active memory [17] since it uses intrinsic processes which allow not only a simple access to the data but also provides intelligent maintenance functionalities. One of the intrinsic processes for example enables the autonomous removal of obsolete information about objects (e.g., the position of a cup three months ago). This *forget* mechanism is quite essential for our application since the robot's environment is continously changing.

Within our work we have extended the functionality of the active memory in order to be able to handle the different modalities by storing the same data in different formats. This information that might seem to be redundant at first glance is necessary because the Scene Model is used as BIRON's long-term memory and both Dialog System and OAS access the data stored in it. For example, consider the color of an object: the Dialog System may store its value in form of a character string, e.g.,"blue"; but after finding it in the current scene the OAS may need to store its color value based on the *Hue Saturation Intensity* (HSI) color model. The same holds true for the position of an object, which the Dialog System would store symbolically as "on the table" while the OAS would store its coordinates. The coordinates describing the position of an object are obviously quite useless for the Dialog System when the user asks "where is my blue cup?". On the other hand, the OAS would not be able to handle the value "blue" when it has to find a blue cup in the camera image.

Consequently, the Scene Model needs to include a component that is able to convert the format of data, the so-called *Modality Converter*. The Modality Converter is a simple yet powerful mechanism that is not only able to convert single object features like the color. It can also search the data base and will return whole object entries matching given descriptions. This may be necessary for example when the OAS fails to detect an object by matching all memorized views against the current camera image and the object's color is not yet known to the OAS. Then, in order to extend the search for visual object features, the OAS sends a request for the object's color to the Dialog System. Subsequently, a search for the newly given color can be performed, after an appropriate conversion of the Dialog System's response is received by the OAS.

For the conversion process the Modality Converter uses a lookup table (cf. Table 1) that contains for every stored *predicate name* (e.g., color, relation, ...) two attribute fields, in particular a *symbolic description* as well as a *visual feature description*. Since the HSI values might vary for a distinctive verbally named color, the corresponding value field can contain specific values as well as ranges of values. Depending by which module a query is sent the Scene Model returns automatically the adequate description if available. For instance, if the query origins from the Dialog System the Converter will automatically return the attribute "Symbolic".

Predicate name	Symbolic	Visual
Color	red	330..20,0..1.0,0..1.0 0.0,1.0,1.0
Color	green	135,1.0,0.7
Relation	O_1 under O_2	$O_1.y < O_2.y$

Table 1: Lookup table of the Modality Converter

Note that this converter is a very powerful tool since it does not only convert pre-defined symbol-value pairs but it is also able to learn new associations. In sum, three different responses to queries are possible. ① The converter finds an entry where all data fields match the attributes of the specified object. In this case, it will return the value suitable for the inquiring component. ② The converter finds no valid entry because there is no correspondence of the entry in the other modality as for example for the symbolic name "transparent" which does not have an HSI equivalent. ③ The converter finds no valid entry because it is not yet complete. This usually occurs when after a search for visual object features a new HSI value is stored in the Scene Model for which no symbolic name is yet known. Then, the Dialog System will ask the user for the color name of the Region Of Interest.

7. RESULTS

In order to illustrate our results we present a dialog example where the resolution of object references is involved. In this example, the user asks the robot to pay attention to a mug. Fig. 6 illustrates the dialog flow, the operations of the modules underlying the robot output and the content of the Scene Model in the first, second and third column respectively. We assume that the robot has already recognized the user as Thomas.

In the utterance U1 the word "this" indicates a possible accompanying gesture which can help to specify its meaning. The Dialog System therefore sends a request to the OAS to search for the object mug (❶). Upon this request the OAS will first query the Scene Model for an object of the type mug in order to provide a template to the object recognizer (❷). In our example, no such object is stored in the Scene Model (❸). The OAS now switches to its second searching strategy: search with salient features of the object in the current scene. But since neither salient fea-

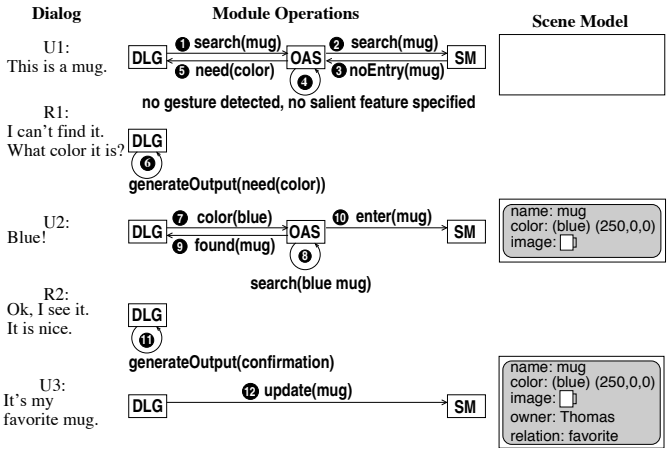


Figure 6: A dialog example (U: User; R: Robot; DLG: the Dialog System; OAS: the Object Attention System; SM: Scene Model)

tures of the mug were specified, nor a gesture was found in the current scene (❹), the OAS informs the Dialog System about its need of a salient feature, namely the color (❺). The Dialog System will then generate the output R1 to get the requested information from the user (❻). Thomas answers the question and the value of the color is sent from the Dialog System to the OAS (❼). With this information the OAS successfully finds the object mug (❽) and informs the Dialog System about this result (❿). At the same time the multi-modal information about the mug is entered in the Scene Model by the OAS (⓫). The Dialog System can now generate a confirmation (⓬) as feedback to the user. In U3 the user specifies two further features of the mug, the owner ("mine") and a description ("my favorite"). This information is directly entered into the Scene Model by the Dialog System since there is no evidence of the involvement of a user gesture, which would indicate that a new object is being specified, and there is only one entry in the Scene Model that matches the described object.

8. CONCLUSION

In order for a mobile robot to be able to communicate in a human-style it needs processing and representation strategies that can deal with multi-modal information. We therefore integrated a Dialog System using multi-modal interaction units with an Object Attention System that is able to resolve object references. The interaction between these modules is based on a multi-modal representation, the Scene Model, which stores the acquired scene information and provides a Modality Converter that not only converts information from one modality to another but also can learn associations between data from different modalities such as color names and HSI values. This powerful mechanism allows on the one hand to use a pre-defined knowledge base while it is on the other hand capable of adapting to new environments by learning new objects and salient features.

The current system still has some limitations. Firstly, the robot cannot yet learn new words, this means, the robot can only learn objects with known symbolic names. A solution of this problem can be adding a mechanism into the DLG that can store and reuse new words once they are spelled by the user. Secondly, a global 3D coordinate system representing the absolute position of an object in relation to the room is not yet integrated into the OAS. The consequence is, the robot can only find the object again if it is in the position as it learned the object. Thirdly, no navigation system is implemented for the robot so that the robot cannot autonomously move from one location to another to find an object. These limitations are also the motivation for our future work.

9. REFERENCES

[1] K. Aoyama and H. Shimomura. Real world speech interaction with a humanoid robot on a layered robot behavior control architecture. In *IEEE Int. Conf. Robotics & Automation*, pages 3825–3830, 2005.

[2] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Mulanda. Humanoid robots as cooperative partners for people. *Int. Journal of Humanoid Robots*, 2004.

[3] J. S. Brunder. *Towards a Theory of Instruction*. Norton, New York, 1966.

[4] J. E. Cahn and S. E. A psychological model of grounding and repair in dialog. In *Proc. Fall 1999 AAAI Symposium on Psychological Models of Communication in Collaborative Systems*, 1999.

[5] H. H. Clark, editor. *Arenas of Language Use*. University of Chicago Press, 1992.

[6] S. S. Ghidary, Y. Nakata, H. Saito, M. Hattori, and T. Takamori. Multi-modal interaction of human and home robot in the context of room map generation. *Autonomous Robots*, pages 169–184, 2002.

[7] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer. A multi-modal object attention system for a mobile robot. In *Proc. Int. Conf. on Intelligent Robots and Systems*, 2005.

[8] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.

[9] J. M. Iverson, O. Capirci, E. Longobardi, and M. C. Caselli. Gesturing in mother-child interactions. *Cognitive Develpment*, 14(1):57–75, 1999.

[10] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer. Supporting advanced interaction capabilities on a mobile robot with a flexible control system. In *Proc. Int. Conf. on Intelligent Robots and Systems*, pages 3649–3655, 2004.

[11] J. P. Lewis. Fast template matching. In *Proc. Conf. on Vision Interface*, pages 120–123, 1995.

[12] D. McNeill. *Hand and Mind: What Gesture Reveal about Thought*. University of Chicago Press, 1992.

[13] P. Prodanov and A. Drygajlo. Decision networks for repair strategies in speech-based interaction with mobile tour-guide robots. In *IEEE Intern. Conf. Robotics & Automation*, pages 3052–3057, 2005.

[14] E. Schegloff and H. Sacks. Opening up closings. *Semiotica*, pages 289–327, 1973.

[15] D. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, University of Rochester, 1994.

[16] D. Traum and P. Dillenbourg. Miscommunication in multi-modal collaboration. In *Proc. AAAI Workshop on Detecting, Repairing, And Preventing Human–Machine Miscommunication*, 1996.

[17] S. Wrede, M. Hanheide, C. Bauckhage, and G. Sagerer. An Active Memory as a Model for Information Fusion. In *Proc. Int. Conf. on Information Fusion*, pages 198–205, 2004.