ELSEVIER

Robotics and Autonomous Systems 43 (2003) 133–147

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Multi-modal anchoring for human–robot interaction

J. Fritsch*, M. Kleinehagenbrock, S. Lang, T. Plötz, G.A. Fink, G. Sagerer

Applied Computer Science, Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany

Abstract

This paper presents a hybrid approach for tracking humans with a mobile robot that integrates face and leg detection results extracted from image and laser range data, respectively. The different percepts are linked to their symbolic counterparts *legs* and *face* by anchors as defined by Coradeschi and Saffiotti [Anchoring symbols to sensor data: preliminary report, in: Proceedings of the Conference of the American Association for Artificial Intelligence, 2000, pp. 129–135]. In order to anchor the composite object *person* we extend the anchoring framework to combine different component anchors belonging to the same person. This allows to deal with perceptual algorithms having different spatio-temporal properties and provides a structured way for integrating anchor data from multiple modalities. An evaluation demonstrates the performance of our approach.

Keywords: Anchoring; Multi-modal person tracking; Human–robot interaction

1. Introduction

The increasing availability of mobile robot platforms with good navigation capabilities provides a basis for the exploration of advanced human–robot interfaces (HRI). The development of systems with natural HRI is an important prerequisite for the widespread use of robots in home and office environments [1]. However, building powerful interfaces that go beyond a simple dialog-based interaction between user and robot is challenging. Due to the nature of mobile systems it is necessary to use sensor devices that can be carried on-board a small robot for realizing an HRI. Additionally, the sensing techniques must be non-intrusive, i.e. a human must be allowed to interact with the robot without having to wear special equipment (e.g. markers, colored gloves) to enable the robot's sensors to observe him. Standard multi-media cameras are cheap sensors that can be used for observing a human instructor to track his position and recognize gestural instructions [3,22]. However, despite intensive research in computer vision, the variations in lighting conditions encountered in dynamic environments pose major problems for tracking humans based on their visual appearance. For example, the color of a human face changes significantly if the lighting conditions are varied. A face detection process based on color may therefore fail to always detect the face in the images of a sequence depicting a human moving through an office. At the same time there may be background objects entering the field of view of the camera that have a face-like color. Consequently, the feature sequence belonging to an image sequence may contain false positives (background objects) and false negatives (missed faces).

In order to enable the robot to track humans over time despite inaccuracies in the feature sequence, the tracking algorithm can make use of temporal information and context knowledge. These sources of

* Corresponding author. Fax: +49-521-106-2992.
E-mail address: jannik@techfak.uni-bielefeld.de (J. Fritsch).

0921-8890/03/$ – see front matter
PII: S0921-8890(02)00355-X

information allow to (1) select the features matching an internal symbolic description of the object to be tracked, and (2) focus processing on a subset of all features. The latter is especially important if the sensor capability is limited, the processing power is small, or several objects are present.

The *anchoring* framework by Coradeschi and Saffiotti [4,5] aims at providing a method for tracking objects over time by defining a theoretical basis for grounding symbols to percepts originating from physical objects. The practical capability is demonstrated with examples dealing with a single type of percept obtained by processing camera images.

However, in complex environments several different sensors can generate different types of percepts originating from the same physical object. Additionally, the spatio-temporal properties of the different types of percepts can vary significantly. We propose a solution to these problems by anchoring a symbol denoting a *composite object* through anchoring the symbols of its corresponding *component objects*. In this solution, the composite anchoring module is responsible for fusing the data of the component anchors. Our approach to integrate several anchoring processes can be easily extended to other modalities and allows for parallel or distributed anchoring of component symbols. To demonstrate our approach we perform person tracking by anchoring the symbol *person* through anchoring the symbols *legs* and *face* to the corresponding percepts.

In extension to the original use of anchoring for connecting one symbol system to one perceptual system, our application concentrates on solving the challenging task of tracking composite objects, i.e. humans. Therefore, we use a symbol system that only contains predicate symbols describing the identity of persons to be tracked. The use of more predicate symbols in the symbol system to support more complex interactions using, for example, speech (e.g. 'Follow the small person with the red shirt') will be the focus of future work.

In the following section we will give a review of related work. The original anchoring framework will be described in Section 3. The basic idea of the proposed integration framework is presented in Section 4, while Section 5 describes some extensions to cope with multiple composite objects. The application to person tracking is described in Section 6. Section 7 presents an extensive evaluation of the complete system. The article concludes with a summary of the presented work.

2. Related Work

Our approach extends work by Coradeschi and Saffiotti [4,5], and therefore their anchoring framework is described in detail in Section 3. In this section we will concentrate on the related techniques of data association and fusion, as these techniques bear similarities to our approach.

Bar-Shalom and Li discuss in [2, Ch. 8.2] different types of configurations for multisensor tracking including a hybrid approach. The so-called Type I configuration denotes a standard single sensor tracking system. Type II configurations perform Type I tracking for several sensors and subsequently fuse the individual tracks, while Type III proposes a direct *synchronous* sensor data fusion across multiple sensors before performing tracking on the fused sensor data. The Type IV configuration, instead, uses local data association for the individual sensors but a global tracking. However, this configuration still requires synchronous sensor data. For fusing data originating from sensors at different sites, a hierarchical hybrid configuration for multisensor–multisite tracking is proposed.

For person tracking using different sensing modalities a variety of approaches and fusion methods have been developed. Darrell et al. [6] use a Type II data fusion method to integrate depth information, color segmentation, and face detection results. Fusing the individual tracks is done using simple rules. Likewise, Okuno et al. [11] use a Type II configuration to fuse auditory and visual information from talking persons. Track fusion is done rule-based, but differently from [6] thresholds on the track differences are used to avoid fusing different tracks. A Type III configuration is used by Feyrer and Zell [7] to track persons based on vision and laser range data. The two types of sensor data are fused by adding a two-dimensional Gaussian to a potential field representation for each potential person position. After initial selection of the person to be tracked, another Gaussian is added to the potential field at the Kalman filtered position estimate to maintain temporal coherence. Type IV configurations with sequential processing of the individual sensors

are often implemented hierarchically. After associating coarse position estimates, a smaller search space is used for processing more precise sensor data. Schlegel et al. [15] propose vision-based person tracking that uses color information to restrict the image area that is processed to find the contour of a human. A more sophisticated method to realize the sequential search space reduction is proposed by Vermaak et al. [21]. In their approach sound and vision data are sequentially fused using particle filtering techniques.

Although we perform person tracking using a camera and a laser range finder which are on-board a mobile robot, we have to perform multisite tracking in a hybrid configuration, as different components of a human are observed from different positions. In contrast to the intersite association and overall information fusion proposed in [2] we developed a model-based modular integration scheme that extends the anchoring framework described in Section 3. Besides enabling classical tracking with multiple sensors at different sites, anchoring allows to maintain representations for temporarily occluded objects and provides mechanisms for reacquiring the object. Therefore, anchoring can be understood as an extension to classical tracking approaches that defines a framework for dealing with missing sensor data in a structured way. The proposed multi-modal anchoring approach is easy to implement, has transparent structure, and exhibits efficient, low complexity performance.

3. Anchoring

The problem of recognizing objects by linking features extracted from sensor data to an internal symbolic representation is especially prominent in an autonomous system whose environment is constantly changing. Such a system needs to establish connections between processes that work on the level of abstract representations of objects in the world (symbolic level) and processes that are responsible for the physical observation of these objects (sensory level). These connections must be dynamic, since the same symbol must be connected to new sensor data every time a new observation of the corresponding object is acquired.

We follow the definition of anchoring proposed by Coradeschi and Saffiotti [5]. They define anchoring as the problem of creating and maintaining in time the correspondence between symbols and sensor data that refer to the same physical object. Basically anchoring incorporates a *symbol system* and a *perceptual system* that are linked by an anchor (Fig. 1). The symbol system includes a set of individual symbols and a set of unary predicate symbols. Each individual symbol has a symbolic description which is a set of predicate symbols. The perceptual system includes a set of *percepts* and a set of *attributes*. A percept is a structured collection of measurements assumed to originate from the same physical object. An attribute is a measurable property of a percept. The set of attribute-value pairs of a percept is called the *perceptual signature*.

The role of anchoring is to establish a correspondence between a symbol, which is used to denote an object in the symbol system, and a percept generated in the perceptual system by the same object. This is achieved by comparing the symbolic description and the perceptual signature via a predicate grounding relation g. This relation constitutes the correspondence between unary predicates and values of measurable attributes. For example, g could specify that a symbol with the predicate *large* corresponds to a percept, if the value of its attribute *size* is above a certain threshold. The relation g can be embedded in a function *match* that evaluates whether a given perceptual signature is consistent to a given symbolic description or not. The

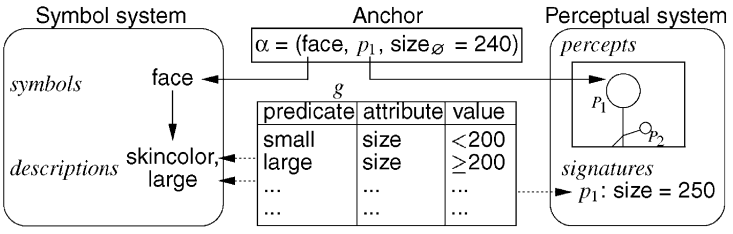

Fig. 1. Linking symbols to sensory data with anchors.

correspondence between symbol and percept is represented in an internal data structure α, called anchor. Since new percepts are generated continuously within the perceptual system, this relation is indexed by time.

At every moment t, the anchor $\alpha(t)$ contains three elements: a symbol, meant to denote an object inside the symbol system; a percept, generated inside the perceptual system by observing an object; a signature, providing the estimate of the values corresponding to the observable properties of the object. The anchor α is *grounded* at time t, if it contains the percept perceived at t and the updated signature. If the object is not observable at t and so the anchor is *ungrounded*, then no percept is stored in the anchor but the signature still provides the best available estimate.

In order to solve the anchoring problem for a given symbol x in a dynamic environment three main functionalities have been outlined in [4,5]:

- *Find*. Create a grounded anchor the first time that the object denoted by x is perceived. The function *match* is used to assure that the symbolic description matches the perceptual signature. In case of multiple matching percepts, a *selection* can either be made inside the find functionality or by the symbol system.
- *Track*. Continuously update the anchor while observing the object. In this case the prediction is achieved by a specific *one-step-predict* function. The predicted signature is compared to the perceived attributes with a *match-signature* function. This allows to find percepts compatible with the attributes of the percepts anchored to the symbol in the previous steps. In case of multiple matching percepts, the *select* function is used to choose one percept.
- *Reacquire*. Update the anchor when the object has to be reacquired, i.e. if the anchor is ungrounded. This is used to locate an object when there is a previous perceptual experience of it. The experience is used to *predict* a new signature which is then compared to newly acquired percepts. Here, the prediction is generally more complex than in the *track* case. If it is *verified* by using *match* that a percept is compatible with the prediction and the symbolic description then the current signature is *updated*. Again, in case of multiple matching percepts, a *select* function is used to choose one percept for updating.

For a detailed description of the formal anchoring framework the interested reader is referred to [4,5].

4. Multi-modal anchoring

Up to now the literature on anchoring considers only the special case of connecting one symbol to the percepts from one sensor. However, the real world contains objects that cannot be captured completely by a single sensor. If several sensors are used, the symbolic description of the object has to be linked to several different types of percepts acquired from different modalities.

One solution is the extension of the anchoring definition to link several percepts to a single symbol. However, with such an approach the integration of different types of percepts with different processing times makes it necessary to anchor the individual percepts asynchronously. Additionally, if the different percepts relate to different parts of the object the spatial relations between them need to be incorporated into the predicate grounding relation to obtain a consistent result. Consequently, the resulting algorithm for this solution may become very complex from an implementational point of view.

Therefore, we propose a modular approach (Fig. 2) that allows to anchor a symbol of a composite object by distributed anchoring of the corresponding component objects based on the related percepts originating from multiple modalities. This modular approach provides a structured way for simple integration of

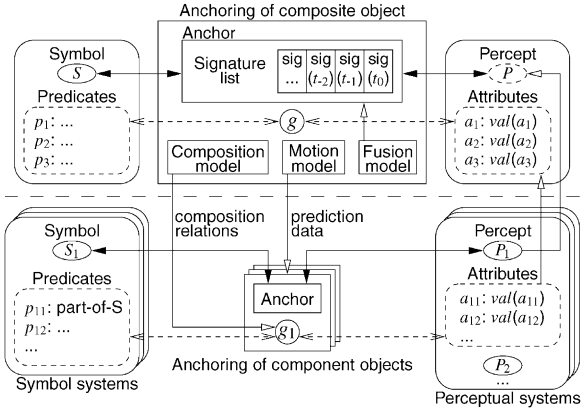

Fig. 2. Multi-modal anchoring.

additional component anchors and facilitates parallel anchoring of different types of percepts. The information provided by the individual perceptual systems of the component anchors is collected by a composite anchoring process for integration. The combined data is again stored in an anchor structure, the so-called composite anchor.

The main difference to original anchoring is that the symbol corresponding to the composite object has no direct perceptual counterpart. Every time a component anchoring process has chosen a new percept for updating its anchor, the percept is linked to the symbol of the composite object. The composite anchoring process then calculates its own perceptual signature by incorporating the signature of the component anchor. Usually, this signature can only be used to update a subset of the available attributes of the composite anchor, because the associated percept originates only from the perceptual system of a component object.

The main functionalities *find*, *track*, and *reacquire* defined in the original anchoring do not directly exist for the composite anchor module. These functions are carried out by the component anchoring processes that also initiate updates of the composite anchor. The composite object is anchored/grounded, if at least one component object is anchored/grounded. Because every component anchor module has different predicate symbols, it also contains its own predicate grounding relation. The predicate grounding relation of the composite anchor module embodies the correspondence between predicates concerning the composite object and attribute data calculated from attribute values originating from the different component anchoring processes.

In order to coordinate all component anchoring processes, it must be ensured that the different sensors observe the same composite object. The component anchoring processes have to be supplied with position estimates for the composite object, and the composite anchoring process has to fuse the information supplied by the component anchors. Therefore, a *composition model*, a *motion model*, and a *fusion model* are provided.

The composition model contains the spatial relationships between the composite object and its components. It ensures that the individual anchoring processes only select percepts that are compatible with the composite object. At startup, a component anchoring process establishes a grounded anchor simply if its symbolic description matches the perceptual signature. Hence, the composite anchor is initialized and from now on data about the composite object is provided to its component anchoring processes as follows: the *match* function of every component anchor is extended to additionally make sure that the composition relations provided by the composition model of the composite object are satisfied. Therefore, the predicate *part-of-S* is added to the symbolic description of the component anchors where *S* is the symbol of the corresponding composite object. After a component anchoring process has executed its extended *match*, the composite anchoring process can perform its own *match* to check whether its symbolic description matches the corresponding perceptual signature of the processed percept.

The motion model describes the motion behavior of the composite object and allows to predict its position. Together with the spatial relations provided by the composition model a component anchoring process can predict the position of its underlying component object. Especially for steerable sensors which allow to select the desired field of view it may be necessary to use information about the composite object. In this case a steerable sensor can be pointed into the direction where a percept is expected in order to establish the corresponding component anchor.

The fusion model is used for integrating the various signatures of the component anchors in the composite anchor. Every time a component anchoring process has processed new percepts, it sends its new signature to the composite anchor module. This signature refers to the point of time in the past when the corresponding sensor data was acquired. Since the different perceptual systems achieve different processing speeds, the composite anchor module does not always receive the attribute data from component anchors in correct temporal order. In order to ensure that the attribute data is fused to the signature of the composite anchor at the appropriate point of time, the composite anchoring process maintains a list containing all signatures sorted in chronological order. New attribute data is inserted in the list and the signature of the composite anchor is updated for the corresponding point of time based on the fusion model. If the list already contains entries that are newer than the inserted one, then the fusion of the signatures of the composite anchor is repeated for

the subsequent points of time. The underlying specification of the fusion itself is domain dependent.

5. Anchoring multiple composite objects

Usually, more than one object has to be tracked simultaneously. Then, several anchoring processes have to be run in parallel to keep track of the different objects. In this case, multi-modal anchoring as described in the previous section may lead to the following conflicts between the individual composite anchoring processes:

- A percept is selected by more than one anchoring process.
- The anchoring processes try to control a steerable sensor contradictorily.

In order to resolve these two problems, a *supervising module* is introduced, which manages all composite anchoring processes. It coordinates the selection of percepts and schedules access to steerable sensors. The supervising module grants access to steerable sensors only to the composite anchoring process which holds the so-called *anchor of interest*. The decision which is the anchor of interest depends on the intended application.

In order to coordinate the selection of percepts the *select* functionalities of the individual component anchor modules have to be modified. These modules no longer select percepts individually. Instead, they assign to every percept a score, which is the higher the better a percept fits the anchor. Based on these scores an overall selection can be performed (Fig. 3). Any possible selection result can be expressed as a list of assignments, where the nth entry of the list contains the number of the percept which is selected for the nth anchor. Note, that the entries of the list have to be pairwise different in order to describe a consistent result. The total score of an overall selection is defined as the sum of the scores corresponding to the assignments. The aim is to find the optimal result, which is the selection yielding the maximum total score. The corresponding search is realized using a search tree: the root of the tree is given by the empty list, whose list entries are all undefined. For every successive node the number of undefined entries decreases by 1. The leaves of the search tree contain all possible overall selections. Since the maximum of all scores assigned by an anchor is known, the total score of a partially undefined list can be estimated optimistically. Hence, the search can be efficiently realized using the A*-algorithm.

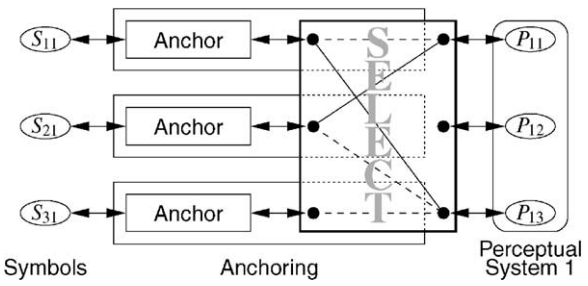

Fig. 3. Modification of *select* in component anchor modules.

However, the number of percepts not necessarily coincides with the number of anchors. If there are more anchors than percepts, not every anchor is assigned a percept and therefore is not updated. If there are more percepts than anchors, not every percept is assigned to an anchor. The remaining percepts are used to establish new anchors. Additionally, an anchor that was not updated for a certain period of time will be removed by the supervising module.

6. Person tracking in a dynamic environment

In order to prove the feasibility of our multi-modal anchoring approach, we demonstrate its use for person tracking with a mobile robot. Person tracking is a prerequisite for every HRI and has to be realized with the available on-board sensors which often can capture only a part of the human body due to the usually small distance between the human and the robot. Our robot can observe a person with a camera and a laser range finder. Based on the skin-colored regions extracted from camera images the face of a person can be detected and identified. The beam from the laser range finder is at leg-height and, consequently, human legs can be detected. In this section we will first present our mobile system. Then, the algorithms to extract the leg and face percepts will be described. Finally, component anchoring and anchoring of the composite object person is explained.

Our hardware platform (Fig. 4) is a PeopleBot from ActivMedia with two on-board PCs (Pentium III, 850

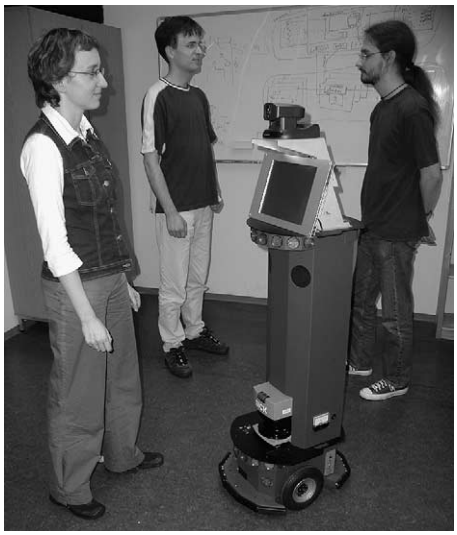

Fig. 4. Our PeopleBot following a person.

and 500 MHz, respectively). The first PC is used for controlling the motors and the on-board sensors while the second one is used for image processing. Both PCs run Linux and are linked with a 100 Mb Ethernet. A SICK laser range finder is mounted at the front at a height of approximately 30 cm. Measurements are taken in a horizontal plane, covering a 180° field of view. A pan-tilt color camera (Sony EVI-D31) is mounted on top of the robot at a height of 140 cm for acquiring images of the upper body part of humans interacting with the robot. For robot navigation we use the ISR (Intelligent Service Robot) control software developed at the Center for Autonomous Systems, KTH, Stockholm [10].

6.1. Detection of human pairs of legs in 2D laser scans

In mobile robotics 2D laser range finders are often used, primarily for robot localization and obstacle avoidance. A laser range finder mounted at the height of legs can also be applied to detect persons. Fig. 5 shows a sample laser scan with a person standing in front of the robot. The legs result in a characteristic pattern.

Detecting legs in laser scans was already considered for mobile systems. In [16] for every object features like diameter, shape, and distance are extracted from the laser scan. Then, fuzzy logic is used to determine which of the objects are pairs of legs. In [17] local minima in the range profile are considered to be pairs of legs. Since other objects (e.g. trash bins) produce patterns similar to persons, additionally moving objects are distinguished from static objects.

Our approach for the detection of human legs is based on laser scans with an angular resolution of 0.5°. Generally, persons can be located by two closely positioned segments. A segment within a laser scan consists of consecutive reading points with similar distance values, which usually result from a smooth

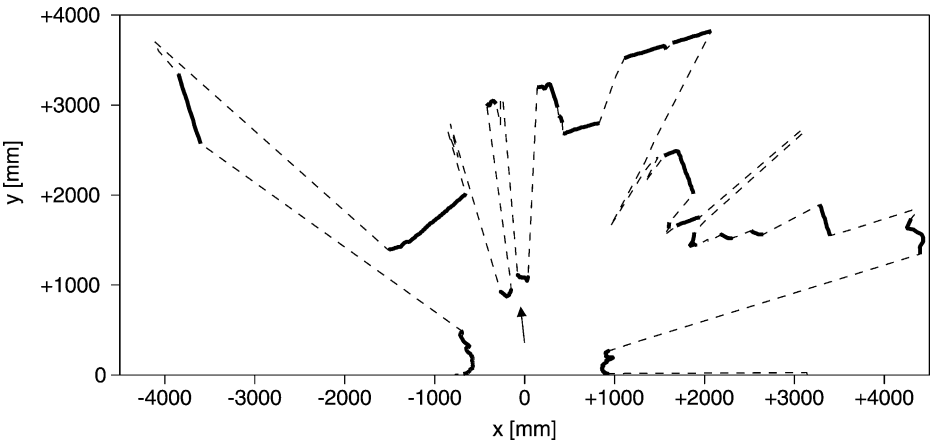

Fig. 5. A sample 2D laser scan. The arrow marks a pair of legs.

surface of a single object. Large differences of distance values are due to edges or occlusions. Thus, single human legs are mostly observed as single segments.

The detection of pairs of legs consists of three steps: *segmentation*, *classification*, and *grouping*. In the first step the laser scan is split into segments. Each segment consists of a maximum number of successive reading points, where the differences of the distance values of two consecutive points are below a given threshold (chosen as 75 mm). In the following step each segment is classified as *leg* or *non-leg*, based on the following features: number of reading points (n), mean distance (μ), standard deviation of the distances (σ), width in world coordinates in a direction perpendicular to the laser beam (w), and distances to the adjacent segments (d_1 and d_2). We obtained satisfying results using the following conditions to classify a segment as *leg*:

$$\begin{aligned}&(n > 4) \wedge (\mu < 3000\,\text{mm}) \wedge (\sigma < 40\,\text{mm})\\ &\quad \wedge (50\,\text{mm} < w < 250\,\text{mm})\\ &\quad \wedge (\max(d_1, d_2) > 250\,\text{mm})\\ &\quad \wedge (\min(d_1, d_2) > -50\,\text{mm}).\end{aligned}$$

In the final step single legs are grouped into pairs depending on their distance in world coordinates, which is chosen to be below 500 mm.

In certain cases one leg of a person is occluded by the other one, and thus only a single leg will be detected. In order not to discard this information, the *percepts* generated by this perceptual subsystem include all detected pairs of legs, and all single legs which are not part of a pair. The *attributes* computed for a percept are the direction given in the local coordinate system of the robot, the distance, and a flag, which indicates whether the percept is a pair of legs or not. The arrow in Fig. 5 marks a pair of legs detected with our approach in the sample laser scan.

6.2. Detection of human faces in color images

Face detection is very important for human–robot interaction: at first, the detection of a face is a reliable indicator for the presence of a person. In addition, much information is extractable from a face, e.g. person identity or gaze direction.

The perceptual subsystem that performs face detection processes color images from the pan-tilt camera mounted on top of the robot. The detection is modeled as an image scanning process, that repeatedly extracts sub-images for classification. To speed up the scanning process, the search space in the image is restricted to regions of skin color. Since we are dealing with images obtained from a camera on a mobile robot, the task of color segmentation is challenging:

- A moving robot encounters lighting conditions of high variability.
- There is no constant background in images as the robot acts in an unstructured environment.

In order to detect color regions under varying lighting conditions, an adaptive color segmentation algorithm has to be used. Probably the most famous adaptive image segmentation system is the Pfinder (person finder) system [23] for tracking a single, completely visible human wearing homogeneously colored clothes in front of a static background. In Pfinder, the color of every background pixel and each body part (head, torso, arms, hands, legs, and feet) is modeled as a Gaussian in YUV color space. Additionally, the positions of body parts are described by Gaussians in image coordinates.

For the task of skin color segmentation the related LAFTER system [12] uses similar techniques to track the face of a single user with a pan-tilt camera. Here a Gaussian mixture is used to model the background variations. In order to detect a face in arbitrary backgrounds captured by a moving camera, recent approaches avoid explicit background modeling [13,18]. However, these approaches are limited to single faces.

Different from the approaches above our goal is the tracking of *several* skin-colored image regions that may be subjected to different lighting conditions. This is realized by modeling every skin-colored region with a separate Gaussian distribution. In order to stabilize the adaptation step, we use context information from face detection to restrict updating to regions containing faces and select image areas of face size for adapting the color models. In the following we give a short overview of our adaptive skin color segmentation approach, more detailed information can be found in [8]. Note, that for skin color segmentation on the mobile robot no region-of-interest (ROI) is used and the complete image is segmented as the uncertainty for determining ROIs on a mobile robot is too high to be reliable enough.

For color representation the r–g color space is used as it is well suited for representing skin color over a wide range of lighting conditions [24]. For the special case of modeling a person's face a Gaussian distribution has been shown to be sufficient [14]. For every pixel the skin probability is calculated as the maximum of the individual probabilities of the Gaussian models. The resulting skin probability image is binarized with an empirically determined threshold of 0.2 and a connected components analysis yields skin-colored regions.

In order to prevent the color models from adapting to skin-colored background objects a face verification step is carried out on all regions found. For face detection we apply the *eigenface method*, operating on gray-level images. Any image with a size of $n \times m$ pixel can be considered as a point in an nm-dimensional space. Faces lie in a subspace of the overall image space. Kirby and Sirovich [9] demonstrated how Principal Component Analysis (PCA) can be used to efficiently represent human faces. Later, Turk and Pentland [20] applied this technique to face detection. PCA finds the principle components of the distribution of the face images, which are called *eigenfaces*. They span a subspace (face space) representing possible face images. We use a face space computed from a set of sample face images having a size of 37×43 pixel. These samples only contain the central parts of faces (eyes, nose and mouth) so that variances of the background are excluded. In addition, the images are preprocessed using histogram equalization in order to compensate varying lighting conditions.

Before a given image can be classified it has to be rescaled to the size of the sample images and preprocessed. The resulting image is then reconstructed by a weighted sum of eigenfaces. The resulting residual error is small if the given image is a face image and large otherwise. Hence, for classification an empirically determined threshold can be used to distinguish face from non-face images.

In order to decide whether a segmented region of skin color originates from a face, a sub-image at the position of the region has to be extracted and classified with the eigenface method. However, the center of mass (COM) of the region does not necessarily coincides with the center of the face due to inaccuracy of segmentation. Therefore, the area at the region has to be scanned at different positions and with varying scalings by using the following method: the center of the initial sub-image (x, y) coincides with the COM of the skin-colored region. There and at the two neighboring positions $(x+1, y)$ and $(x, y+1)$ the corresponding reconstruction errors for the extracted sub-images are computed. The next position of the scanning process is chosen according to *steepest gradient descent*. This process stops if a face is detected or a local minimum is reached. In the latter case the process continues with sub-images of a new size ($\pm 7.5\%$, followed by $\pm 15\%$).

For all image regions that are found to contain a face updating of the color model is performed. In order to stabilize the updating process an empirically determined global skin color distribution is used for filtering out non-skin pixels. Based on a theoretical model Störring et al. [19] have shown that the overall skin color distribution occupies a shell-shaped area in r–g color space that is called *skin locus*. Similar to Soriano et al. [18] we determined the skin locus for our camera empirically with hand-segmented training images [8]. With all pixels in an elliptical image area at a detected face position that lie inside the skin locus, local Gaussian parameters are calculated and used to smoothly update the Gaussian model:

$$\vec{\mu}_{\text{new}} = \gamma \vec{\mu}_{\text{local}} + (1-\gamma)\vec{\mu}_{\text{old}},$$
$$\mathbf{\Sigma}_{\text{new}} = \gamma \mathbf{\Sigma}_{\text{local}} + (1-\gamma)\mathbf{\Sigma}_{\text{old}}.$$

For our system running at approximately 3 Hz a learning rate of $\gamma = 0.6$ has been shown to provide good results for persons moving in a standard office domain.

The *percepts* generated by this perceptual subsystem are the skin-colored regions classified as face. For every percept a set of *attributes* is computed: with the position information from the pan-tilt camera the angle of the face relative to the robot is calculated. The detected face size is used to estimate the distance d of the person: assuming that sizes of heads of adult humans only vary to a minor degree, the distance is proportional to the reciprocal of the size. The height of the face above the ground is also extracted by using the distance and the camera position.

Additionally, a face identification step is performed with a slightly enhanced version of the method proposed in [20]. Each individual is represented in face

space by a mixture of several Gaussians with diagonal covariances. Practical experiments have shown that the use of 4–6 Gaussians leads to satisfying results in discrimination accuracy requiring only small amounts of training material. The mixture densities are estimated from the projections of up to 50 sample images per individual. The performance of the identification process has been evaluated in an experiment with nine individuals. For a test set of 76 images a recognition rate of 89% could be achieved. When accepting a rejection rate of 20%, over 96% of the images classified were assigned to the correct individual.

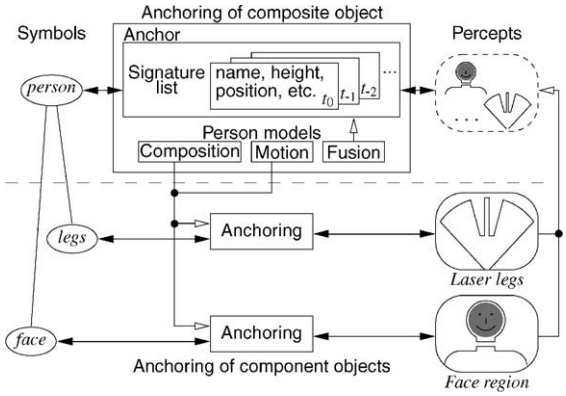

Fig. 6. Anchoring a person by anchoring the legs and the face.

6.3. Anchoring component objects

The characteristics of the anchoring processes for the components legs and face are reflected in their three main functionalities. The *find* functions of the leg and face anchor modules anchor only percepts in front of the robot, if their distance to the robot is less than 3 m. Additionally, the selected leg percept must match the predicate *is-pair*. If the face anchor module is linked to the anchor of interest, it is first checked in the *find* function whether the field of view of the camera overlaps with the person position provided by the anchor of interest. If necessary, the camera is pointed to the direction where the face percept is expected. The functionalities *track* and *reacquire* of the anchor modules for legs and face are rather similar. All these functions try to anchor percepts close to the predicted position while considering restrictions given by the composition model of the person. More specifically, the *track* functions predict the current percept's position based on the last known position. In contrast, the prediction of the *reacquire* functions is based on the person position obtained from the person anchor module. If the face anchoring process tracks or tries to reacquire the face of the person of interest, the camera is steered to make sure that the position of the predicted percept does not move out of the field of view.

6.4. Anchoring composite objects

The person anchoring module receives individual signatures originating from the leg and the face anchoring processes. It is important to note that this data is processed asynchronously by the composite anchoring process. Fig. 6 shows the framework for anchoring the composite object *person*. The *composition model* used describes empirically defined person relations (Fig. 7).

All attributes of the multi-modal anchoring of persons that correspond to spatial positions are described by Gaussian distributions instead of scalar values. This allows to model uncertainty for positions. For the attributes of percepts the variance of the Gaussian can be determined from the measuring inaccuracy of the corresponding sensors. The *motion model* defines how a position can be predicted for time $t(i+1)$ based on the known position at time $t(i)$: the mean value remains unchanged (no velocity assumed) while the variance increases linearly with time, expressing increasing uncertainty.

The attribute values contained in the signature list of the composite anchor module are updated by multiplying the Gaussian of each attribute value with the Gaussian representation of the corresponding attribute values from new percepts. This results in the following update formulas that are calculated in the *fusion*

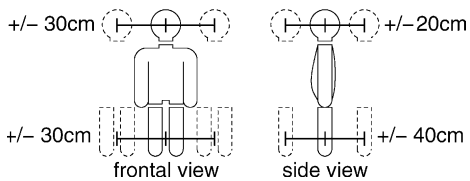

Fig. 7. The composition model for matching consistent percepts.

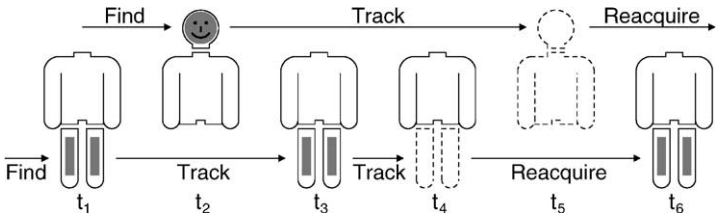

Fig. 8. A schematic example for anchoring a person.

model:

$$\mu_{t(i)} = \frac{\mu_{t(i-1)}\sigma_{\mathrm{p}} + \mu_{\mathrm{p}}\sigma_{t(i-1)}}{\sigma_{t(i-1)} + \sigma_{\mathrm{p}}},$$

$$\sigma_{t(i)} = \frac{\sigma_{t(i-1)}\sigma_{\mathrm{p}}}{\sigma_{t(i-1)} + \sigma_{\mathrm{p}}}.$$

The resulting mean value $\mu_{t(i)}$ is a weighted sum of the old mean value $\mu_{t(i-1)}$ and the mean value of the percept μ_{p}. Since the weights are given by the variances of the old position in the signature list and the percept, the mean value corresponding to the smaller variance (more certainty) has a greater effect.

The person attribute values that are updated with the signatures of the grounded component anchors are the angle ϕ_{p} and distance d_{p} relative to the robot, the face height h_{p} and the person name N_{p}. The initialization of the values ϕ_{p} and d_{p} is carried out if a component anchor is grounded for the first time. The attribute values h_{p} and N_{p} can only be initialized after receiving the first signature from the face anchoring process. During normal operation the person's fusion model makes sure that the person's position is smoothly updated by anchored legs and faces. In contrast, h_{p} and N_{p} can only be updated by processing face signatures.

In order to illustrate the concept a schematic example for anchoring one person is shown in Fig. 8 depicting six consecutive time steps at the beginning of an anchoring process:

t_1 : Person anchoring is started and all component anchoring processes perform their *find*. The leg detection generates a leg percept and the legs are anchored for the first time. The leg anchoring process switches from *find* to *track*. Subsequently, the person position contained in the composite anchor module is initialized and the person becomes grounded. Now, the *find* of the face anchor module is able to point the camera into the right direction.

t_2 : The face detection generates a face percept and the face anchor becomes grounded. The face anchoring process switches from *find* to *track* and the person anchor is updated accordingly.

t_3 : Again, the leg detection generates a leg percept. Based on the *track* function, the leg anchor as well as the person anchor are updated.

t_4 : In this time step, new laser range data is processed but no matching leg percept is found by the leg anchoring process. Therefore, it switches from *track* to *reacquire*. No updating of the person anchor takes place.

t_5 : A new camera image is processed but no face percept matching the prediction of the person position is found. Thus, the face anchoring process also switches from *track* to *reacquire*. Now the person is ungrounded since neither the legs nor the face are grounded.

t_6 : In the new laser range data a leg percept matching the predicted person position is found. Now the legs as well as the person are grounded again.

7. Results

We implemented the extended anchoring framework in an object-oriented manner using C++ and added the person tracking functionality to the ISR software [10] on the behavior level. When the robot is instructed to track persons the tracking behavior is started in parallel with other behaviors necessary, for example, obstacle avoidance. The tracking behavior initializes the person anchoring process.

The evaluation of our system was carried out in an office room, more specifically in an area having a size of approximately 4.60 m × 3.40 m. The room was equipped with wooden furniture, which was

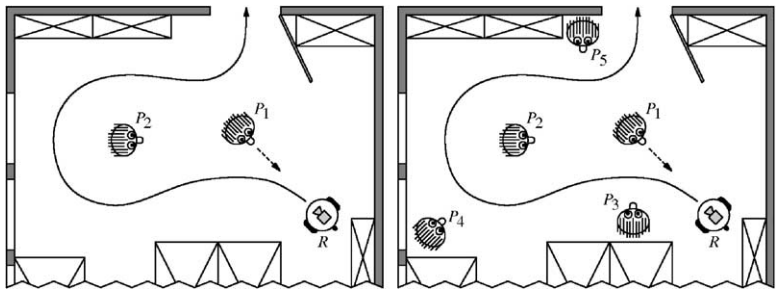

Fig. 9. Scenario: first setup (left); second setup (right).

challenging for the face recognition, because the color of wood is similar to skin color. We realized two setups (Fig. 9). In the first setup only two persons were present, one in the middle of the room standing still (P_2) and one guiding the robot (P_1). The task for P_1 was to become the person of interest by approaching the robot (<1 m). Then, P_1 had to guide the robot around P_2 and to leave the room through the door, while looking towards the camera as long as possible. The resulting trajectory had a length of approximately 7.5 m. The second setup was similar to the first one, but three additional persons P_3–P_5 were placed at predetermined locations in the room, not affecting the trajectory resulting from the first setup. P_1 was instructed to try to regain the attention of the robot in case that the robot lost P_1. If this was not possible, because the robot tried to follow one of the other persons, then the experiment was interpreted as failure. Both experiments were carried out with ten different subjects.

Throughout the tests, the laser range finder provided new laser range data at a rate of 4.6 Hz to the leg detection algorithm. The processing time necessary for generating leg percepts and anchoring was negligible. The adaptive skin-color segmentation processed images with a size of 189×139 pixels. For each skin-colored region the face detection was carried out. The processing time of the face detection and identification system depends on the number of skin-colored regions present in the image. On average the face percepts were provided at a rate of 3.1 Hz. Again, the time necessary for updating component and composite anchor was negligible. Together, the person attributes were updated with an average rate of 7.7 Hz due to the asynchronous anchoring of the different types of percepts.

The first setup (Table 1) was accomplished after an average time of 55 s. The robot lost three people once, but they were able to regain the attention of the robot to complete the run. On average 95.3% of the time a person was grounded. The legs were grounded 92.1%

Table 1
Results of the first setup with P_1 and P_2

Run	t (s)	$v_\varnothing$ (m/s)	Lost	Person grounded (%)	Legs grounded (%)	Face grounded (%)	Legs/step	Face/step
1	39	0.19	0	99.7	98.9	63.1	1.78	0.76
2	62	0.12	0	96.6	93.8	36.4	1.71	0.90
3	52	0.14	1	95.4	83.5	51.0	1.72	0.57
4	56	0.13	0	99.3	93.8	54.1	1.79	0.59
5	81	0.09	1	96.4	95.7	34.7	1.63	0.40
6	32	0.23	0	99.2	98.7	51.1	1.87	0.78
7	90	0.08	1	80.9	73.2	22.0	1.94	0.49
8	51	0.15	0	99.2	98.8	56.5	1.75	0.70
9	42	0.18	0	98.0	97.9	35.7	1.79	0.45
10	44	0.17	0	88.6	87.1	16.3	1.60	0.39
Average	55	0.14	–	95.3	92.1	42.1	1.76	0.60

Table 2
Results of the second setup with P_1–P_5

Run	t (s)	v_ϕ (m/s)	Lost	Person grounded (%)	Legs grounded (%)	Face grounded (%)	Legs/step	Face/step
1	60	0.13	2	93.6	91.5	27.7	2.63	0.41
2	43	0.17	0	96.7	95.0	20.7	2.61	0.32
3	The robot lost P_1 and tried to follow P_3							
4	51	0.15	0	98.7	90.4	66.0	2.49	0.74
5	47	0.16	0	96.2	94.5	7.1	2.52	0.20
6	The robot lost P_1 and tried to follow P_2							
7	77	0.10	0	99.8	97.5	72.0	2.59	0.85
8	74	0.10	0	93.4	92.6	20.3	2.63	0.22
9	61	0.12	0	97.7	96.1	36.4	2.55	0.56
10	42	0.18	0	86.1	84.2	11.9	2.73	0.26
Average	57	0.13	–	95.3	92.7	32.8	2.59	0.45

of the time, the face 42.1%. On average 1.76 legs and 0.6 faces were processed in every computation step by the corresponding perceptual systems.

The time needed to successfully perform the task of the second, more complex setup (Table 2) took only 2 s more per run on average. For this setup we expected more percepts to be computed, because more persons were present. This was in fact true for the legs, but not for the face. The persons guiding the robot were taking care of not colliding with one of the persons P_2–P_5 and, therefore, looked at the camera less often. This resulted in a correspondingly lower face detection rate. On average the face was grounded only 32.8% of the time. The legs and the whole person were grounded for approximately the same time (95.3 and 92.7%) as in the first setup. Runs 3 and 6 resulted in a failure. A recovery was not possible even though the face identification would have indicated the mistake. This is because an active search for a specific person, which goes beyond the reacquire functionality of anchoring, is not part of the current implementation.

8. Summary

We presented a method for anchoring composite symbols through anchoring component symbols to their associated percepts and subsequently fusing the resulting data of the component anchors. This modular approach facilitates multi-modal anchoring and can easily be extended with additional anchoring processes. We demonstrated the performance of our approach with a person tracking application for a mobile robot. In the current implementation laser range data and color images are processed to find percepts for the symbols *legs* and *face*. Our extended anchoring framework allows for multi-modal tracking of humans. Through taking advantage of the different sensor capabilities in terms of precision and information content a more complete representation of tracked persons is maintained. Therefore, our approach forms the basis for more advanced human–robot interaction.

Acknowledgements

This work has been supported by the German Research Foundation within the Collaborative Research Center ‘Situated Artificial Communicators’ and the Graduate Programs ‘Task Oriented Communication’ and ‘Strategies and Optimization of Behavior’.

References

[1] A. Agah, Human interactions with intelligent systems: research taxonomy, Computers and Electrical Engineering 27 (1) (2000) 71–107.
[2] Y. Bar-Shalom, X. Li, Multitarget–Multisensor Tracking: Principles and Techniques, YBS, Storrs, CT, 1995.
[3] H.-J. Böhme, U.-D. Braumann, A. Brakensiek, A. Corradini, M. Krabbes, H.-M. Gross, User localisation for visually based human–machine interaction, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 486–491.

[4] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the Conference of the American Association for Artificial Intelligence, 2000, pp. 129–135.
[5] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the International Conference on Artificial Intelligence, 2001, pp. 407–412.
[6] T. Darrell, G. Gordon, M. Harville, J. Woodfill, Integrated person tracking using stereo, color, and pattern detection, International Journal of Computer Vision 37 (2) (1998) 175–185.
[7] St. Feyrer, A. Zell, Robust real-time pursuit of persons with a mobile robot using multisensor fusion, in: Proceedings of the International Conference on Intelligent Autonomous Systems, Venice, 2000, pp. 710–715.
[8] J. Fritsch, S. Lang, M. Kleinehagenbrock, G.A. Fink, G. Sagerer, Improving adaptive skin color segmentation by incorporating results from face detection, in: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), Berlin, Germany, September 2002, IEEE, pp. 337–343.
[9] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1) (1990) 103–108.
[10] M. Lindstrom, M. Andersson, A. Orebäck, H.I. Christensen, ISR: an intelligent service robot, in: H.I. Christensen, H. Bunke, H. Noltmeier (Eds.), Sensor Based Intelligent Robots, Proceedings of the International Workshop on Sensor Based Intelligent Robots, Selected Papers, vol. 1724, Dagstuhl Castle, Germany, September–October 1998, Lecture Notes in Computer Science, Springer, New York, 1999, pp. 287–310.
[11] H.G. Okuno, K. Nakadai, K. Hidai, H. Mizoguchi, H. Kitano, Human–robot interaction through real-time auditory and visual multiple-talker tracking, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Maui, HI, 2001, pp. 1402–1409.
[12] N. Oliver, A. Pentland, F. Berard, LAFTER: a real-time face and lips tracker with facial expression recognition, Pattern Recognition 33 (2000) 1369–1382.
[13] Y. Raja, S.J. McKenna, S. Gong, Colour model selection and adaptation in dynamic scenes, in: Proceedings of the European Conference on Computer Vision, Freiburg, Germany, 1998, pp. 460–474.
[14] Y. Raja, S.J. McKenna, S. Gong, Segmentation and tracking using colour mixture models, in: Proceedings of the Asian Conference on Computer Vision, vol. 1, Hong Kong, 1998, pp. 607–614.
[15] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, R. Wörz, Vision based person tracking with a mobile robot, in: Proceedings of the British Machine Vision Conference, Southampton, UK, 1998, pp. 418–427.
[16] R.D. Schraft, B. Graf, A. Traub, D. John, A mobile robot platform for assistance and entertainment, Industrial Robot 28 (1) (2001) 29–34.
[17] D. Schulz, W. Burgard, D. Fox, A.B. Cremers, Tracking multiple moving objects with a mobile robot, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, vol. 1, Kauwai, HI, 2001, pp. 371–377.
[18] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, Skin detection in video under changing illumination conditions, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, vol. 1, Barcelona, Spain, 2000, pp. 839–842.
[19] M. Störring, H.J. Andersen, E. Granum, Physics-based modelling of human skin colour under mixed illuminants, Robotics and Autonomous Systems 35 (3–4) (2001) 131–142.
[20] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuro Science 3 (1) (1991) 71–86.
[21] J. Vermaak, A. Blake, M. Gangnet, P. Perez, Sequential Monte Carlo fusion of sound and vision for speaker tracking, in: Proceedings of the International Conference on Computer Vision, vol. 1, 2001, pp. 741–746.
[22] S. Waldherr, S. Thrun, R. Romero, A gesture based interface for human–robot interaction, Autonomous Robots 9 (2) (2000) 151–173.
[23] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[24] J. Yang, W. Lu, A. Waibel, Skin-color modeling and adaptation, in: Proceedings of the Asian Conference on Computer Vision, vol. 2, Hong Kong, 1998, pp. 687-694.

J. Fritsch received the Diploma in Electrical Engineering from the Ruhr-University Bochum, Germany, in 1996. He joined the research group for Applied Computer Science at Bielefeld University, Germany, in 1998. There he is working in the Collaborative Research Center 'Situated Artificial Communicators'. His research interests are adaptive color segmentation, the recognition of manipulation actions based on symbolic and sensory data, and the realization of advanced interfaces for human–machine interaction.

M. Kleinehagenbrock received the Diploma in Computer Science from the RW-TH Aachen, Germany, in 2001. He joined the Research Group for Applied Computer Science at Bielefeld University, Germany, in 2001, as a Ph.D. student in the graduate program 'Strategies and Optimisation of Behavior'. His primary interest is the integration of vision and speech modules on a mobile system to realize an advanced human–robot interface.

S. Lang received the Diploma in Computer Science from University of Bielefeld, Germany, in 2000. He is currently pursuing a Ph.D. program in Computer Science at the University of Bielefeld in joint affiliation with the Applied Computer Science Group and the graduate program 'Task-oriented Communication'. Sebastian Lang is interested in image processing, pattern recognition and human–machine interaction.

T. Plötz received the Diploma in Technical Computer Science from the University of Cooperative Education, Mosbach, Germany, in 1998. In 2001 he received the Diploma in Computer Science from Bielefeld University, Germany. He joined the Research Group for Applied Computer Science at Bielefeld University, Germany, in 2001. Thomas Plötz is interested in HMM-based pattern recognition in the field of speech-processing and bioinformatics.

G.A. Fink received the Diploma in Computer Science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1991 and the Ph.D. degree (Dr.-Ing.) also in Computer Science from Bielefeld University, Germany, in 1995. In 2002 he received the Venia Legendi (Habilitation) in Applied Computer Science from the Faculty of Technology of Bielefeld University. In 1991 he joined the Applied Computer Science Group at the Faculty of Technology of Bielefeld University where he is currently an Assistant Lecturer. His fields of research are speech and handwriting recognition, spoken language understanding, man–machine interaction, and distributed systems for pattern analysis applications. He has published various papers in these fields, and is author of a book on the integration of speech recognition and understanding. Dr. Fink is a Member of the Institute of Electrical and Electronics Engineers (IEEE).

G. Sagerer received the Diploma and the Ph.D. (Dr.-Ing.) Degree in Computer Science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1980 and 1985, respectively. In 1990 he received the Venia Legendi (Habilitation) in Computer Science from the Technical Faculty of this university. From 1980 to 1990 he was with the Research Group for Pattern Recognition at the University of Erlangen-Nürnberg, Erlangen, Germany. Since 1990 he is a Professor of computer science at the University of Bielefeld, Germany, and Head of the Research Group for Applied Computer Science. His fields of research are image and speech understanding including artificial intelligence techniques and the application of pattern understanding methods to natural science domains. He is author, coauthor, or editor of several books and technical articles. Dr. Sagerer is a Member of the German Computer Society (GI), the European Society for Signal Processing (EURASIP) and the Institute of Electrical and Electronics Engineers (IEEE).