Person Tracking with a Mobile Robot based on Multi-Modal Anchoring

M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lömker, G. A. Fink and G. Sagerer
Faculty of Technology, Bielefeld University, 33594 Bielefeld
E-mail: {mkleineh, slang, jannik, floemker, gernot, sagerer}@techfak.uni-bielefeld.de

Abstract

The ability to robustly track a person is an important prerequisite for human-robot-interaction. This paper presents a hybrid approach for integrating vision and laser range data to track a human. The legs of a person can be extracted from laser range data while skin-colored faces are detectable in camera images showing the upper body part of a person. As these algorithms provide different percepts originating from the same person, the perceptual results have to be combined. We link the percepts to their symbolic counterparts legs *and* face *by anchoring processes as defined by Coradeschi and Saffiotti. To anchor the composite symbol* person *we extend the anchoring framework with a fusion module integrating the individual anchors. This allows to deal with perceptual algorithms having different spatio-temporal properties and provides a structured way for integrating anchors from multiple modalities. An example with a mobile robot tracking a person demonstrates the performance of our approach.*

I. 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 [2]. However, building powerful interfaces that go beyond a simple dialog-based interaction between user and system is challenging. Due to the nature of mobile systems it is necessary to use sensor devices that can be carried onboard a small robot for realizing an HRI. Additionally, the sensing techniques must be non-intrusive, i.e. the human must be allowed to interact with the robot without having to wear special equipment to enable the robot's sensors to observe him (e.g. markers, colored gloves). Standard multimedia cameras are cheap sensors that can be used for observing a human instructor to track his position and recognize gestural instructions [3], [14]. However, despite of intensive research in computer vision, the variations in lighting conditions encountered in dynamic environments pose major problems for tracking a human based on the 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).

To enable the robot to track the human over time despite of inaccuracies in the feature sequence, the tracking algorithm can make use of temporal information and context knowledge. These sources of information allow to I) select the features matching an internal symbolic description of the object to be tracked and II) 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 'interesting' objects are present.

The *anchoring* framework by Coradeschi and Saffiotti aims at providing a method for tracking objects over time by defining a theoretical basis for grounding symbols to percepts originating from physical objects [4], [5]. The practical capability is demonstrated with examples dealing with a single type of percepts 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 symbols denoting *composite objects* through anchoring the *component symbols* they are comprised of and 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 its com-

*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'.

In Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN), Pages 423-429, Berlin, Germany, September 2002. IEEE.

ponent symbols *legs* and *face*. The use of this model-based method for data fusion improves the tracking of a human in a dynamic environment typically encountered by a mobile robot.

For fusing different sensing modalities a variety of approaches tailored to specific applications have been developed. In sensor-based fusion methods Kalman filtering and more recently particle filtering (see e.g. [11], [13]) are prominent techniques. For the task of multi-modal person tracking Feyrer and Zell [7] use a potential field for performing sensor-based fusion of vision and laser range data. The opposite to sensor-based approaches form rule-based fusion methods where results of individual algorithms are fused based on combination rules (see e.g. [6], [9]). In relation to these two extremes our extension to the anchoring framework by fusing individual anchors forms a hybrid approach.

We start with a description of our mobile robot in section II followed by a review of the anchoring framework in section III. The basic idea of the proposed integration framework is presented in section IV and the application to person tracking based on laser and vision data is described in section V. Section VI gives implementational details and provides a performance example of the complete system. The article ends with a summary of the presented work.

II. Mobile Platform

Our hardware platform is a Pioneer PeopleBot from ActivMedia with an onboard PC (Pentium III, 850 MHz) for controlling the motors and the onboard sensors (Fig. 1). The 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. The 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. We installed an additional PC (Pentium III, 500 MHz) inside the robot in order to enable image processing directly on the mobile platform. The two PC's running under Linux are linked with a 100 Mbit Ethernet and the controller PC is equipped with wireless ethernet to enable remote control of the mobile robot. For robot navigation we use the ISR (Intelligent Service Robot) control software developed at the Center for Autonomous Systems, KTH, Stockholm [10].

III. 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.

Fig. 1. Our PeopleBot following a person

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 percepts everytime a new observation of the corresponding object is acquired.

We follow the definition of anchoring proposed by Coradeschi and Saffiotti in [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* (Fig. 2). 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 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 *small* corresponds to a percept, if the value of its attribute *size* is below 200. The correspondence between symbol and percept is represented in an internal

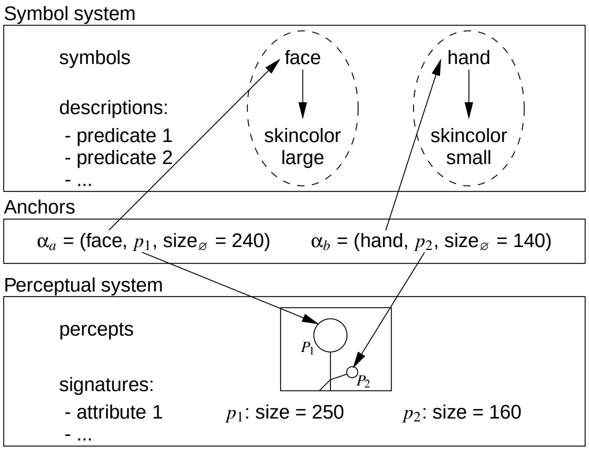

Fig. 2. Linking symbols to sensory data with anchors

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; and a signature, meant to provide the estimate of the values of 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 contained in the anchor but the signature still contains 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] and [5], respectively:

- **Find:** Create a grounded anchor the first time that the object denoted by x is perceived. The grounding relation g 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.
- **Reacquire:** Update the anchor when the object has to be reacquired after some time that it has not been observed. This is used to locate an object when there is a previous perceptual experience of it. This experience is used to *predict* a new signature which is then compared to newly acquired percepts. If it is *verified* that a percept is compatible with the prediction and the symbolic description, again by considering g, then the current signature is *updated*. In case of multiple matching percepts, a *select* function is used to choose one percept for updating.
- **Track:** This special case of reacquisition continuously updates the anchor while observing the object. Consequently, prediction in this case is much simpler than in the Reacquire case, it 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. Again, in case of multiple matching percepts, the *select* function is used to choose one percept.

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

IV. 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 the percepts of 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 anchor 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 requires either synchronization of the percepts or asynchronous anchoring of the individual percepts. Another difficulty emerges if the different percepts relate to different parts of the object. In this case the spatial relations between the different percepts would need to be incorporated into the grounding relation to obtain a consistent result. Together with different temporal properties of the percepts the resulting algorithm for anchoring a composite symbol based on component percepts may become very complex from an implementational point of view.

Therefore, we propose a modular approach that allows to anchor a composite symbol by distributed anchoring of the components based on the related percepts coming from multiple modalities. The information provided by the individual anchoring processes is sent to an Anchor Fusion (AF) module integrating the different component anchors belonging to the composite object (Fig. 3). This modular approach provides a structured way for simple integration of additional component anchors and facilitates parallel anchoring of different types of percepts.

The AF module controls the initialization and termination of the basic anchoring processes. Initialization can be performed on request from the symbol system or on startup if the system is intended to wait for the first occurrence of a certain object. Termination is either caused by a command from the symbol system or based on a timeout if none of the component symbols was successfully grounded for a certain period of time.

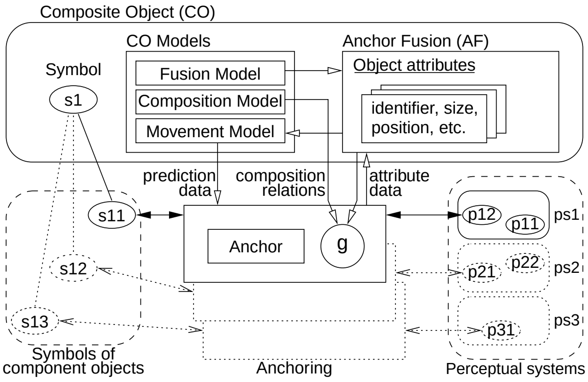

Fig. 3. Anchoring a composite symbol by fusing the anchors of its component symbols

Each time a component anchor has processed new percepts, it sends its new *attribute data* to the AF module. This attribute data refers to the point of time in the past when the corresponding sensor data was acquired. Because the different perceptual systems need different amounts of time to calculate the percepts, the AF module does not always receive the attribute data of the different anchors in correct temporal order. To assure that the attribute data is fused to the *attributes of the composite object* at the appropriate point of time, the AF module maintains a list containing all attribute data sent to the AF module sorted in chronological order. New attribute data is inserted in the list and the attributes of the composite object are 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 attributes of the composite object are again fused for the subsequent points of time. The fusion is realized by calculating a weighted average over the new attribute data and the attributes of the composite object. The weighting of the attribute data depends on the quality of the corresponding perceptual system.

The attributes of the composite object can be used by the component anchors to predict their signature. The composite object supplies a *movement model* to predict the position of the composite object for the current point of time. At the moment the predicted position is simply the position provided by the AF module which is only updated if new data is sent to the AF module from the anchors. The grounding relation g of each anchor is extended to not only check that the symbolic description corresponds to the perceptual signature, but also to make sure that the *composition relations* provided by the *composition model* of the composite object are satisfied if the composite object is already initialized. This ensures that the individual anchors only select percepts that are compatible with the overall composite object.

Special attention has to be paid to the *Find* functions of the component anchors, as in these functions the dependency between the individual component symbols and the composite symbol can be used to control the initialization of the component anchors. Certain anchors may start the *Find* functionality only after initial object attributes are available from the AF module, i.e. another component anchor was successfully grounded. For example, only a spatial information allows to control a camera with a limited field of view to point in the direction where a matching percept is expected. The feasibility of our AF framework is demonstrated in the following sections with a person tracking application for a mobile robot.

V. Person Tracking in a dynamic Environment

With the progress in mobile autonomous systems the development of advanced human-robot-interfaces gains increased attention. However, the prerequisite for any interface is to be aware of the human user and focus its attention towards him. This tracking capability must be robust to movements of the mobile robot and the human and the accompanying variations in the appearance of the human. Additionally, the tracking has to be realized with the available onboard 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 scanner. 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 laser range data. In this section we will first present the anchoring of the individual percepts, before the fusion module for anchoring an entire person is explained.

A. Anchoring legs

We use a 2D laser range finder to detect human pairs of legs. A laserscan consists of 361 reading points covering a 180° field of view. Figure 4 depicts a sample laserscan with a person situated in front of the robot. In order to detect legs neighboring reading points of the laserscan are grouped into segments. Then, each segment is classified as *leg* or *non-leg* according to a set of thresholds. In the final step detected legs are grouped into pairs depending on their distance in world coordinates.

Percepts generated by this perceptual subsystem consist of all detected pairs of legs, and all single legs, which do not belong to a pair. The *attributes* computed for one percept are the direction and the distance given in the local coordinate system of the robot. The arrow in Figure 4 marks the pair of legs detected in the sample laserscan. Given the percepts for legs extracted from laser range data, the an-

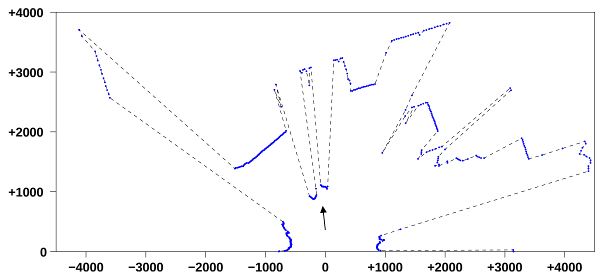

Fig. 4. Sample 2D laserscan. The arrow marks the pair of legs of a person standing in front of the robot.

choring functions for the elementary symbol *legs* are implemented as follows:

- **Find:** Anchor only percepts in a 60° angle in front of the robot at a distance of 150 cm ± 50 cm.
- **Track:** Predict the current leg position (angle and distance) based on the last leg position and the person position, which is provided by the AF module of the composite object. Choose percepts that are consistent with the predicted position and the composition model of the person (see Fig. 5). Then, select the percept closest to the predicted position.
- **Reacquire:** This is the same as in Track except that the current position is predicted based only on the person position. This prediction is received from the movement model of the composite object.

Each time the Leg-Anchor (LA) is updated with a legs percept the attribute data is sent to the AF module for updating the person attributes.

B. Anchoring faces

Face detection is very important for human-robot-interaction: A detected 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 subsystem which generates face percepts performs face detection in two sequential steps. First, the camera image is segmented based on an adaptive skin color segmentation method. Then, every skin colored region is tested, whether it originates from a frontal view face or not. Therefore, at the center of every region sub-images are extracted from the corresponding graylevel image and classified as *face* or *non-face*. A detailed description of this subsystem can be found in [8]. Subsequently, an improved version of the method proposed in [12] is used for face identification. By incorporating the position of the pan-tilt camera and the camera height, the *attributes* provided by the face detection are the angle, the distance and the height of a face in robot coordinates. The face identification provides the associated name of the person.

For the symbol *face* the processing of the face percepts by the anchoring functions is summarized below:

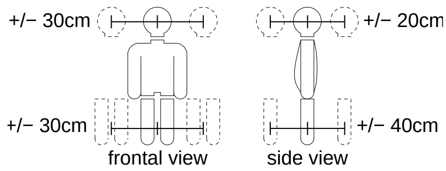

Fig. 5. The composition model for matching consistent percepts

- **Find:** This function waits for an initial person position in the person attributes, i.e. for the legs to be anchored. Subsequently, the camera is directed to point in the direction of the person's position and the image processing is started. Through the composition relations (see Fig. 5) only a face percept at a position close to the person's position is anchored.
- **Track:** Predict the face position based on the last face position and the person position from the AF module. Percepts within a small radius around the predicted position are chosen in the *verify* step if they are also consistent with the person's composition model. Finally, the best match is selected for anchoring.
- **Reacquire:** Here the person position is directly used to be the predicted face position. This data is supplied by the person's movement model. Selection and verification is essentially the same as in track.

Each time the Face-Anchor (FA) is updated with a face percept the attribute data is sent to the AF module for updating the person attributes.

C. Updating the person attributes

Each anchor data sent to the AF module from the individual anchoring processes contains status information about the current anchoring mode (*Find*, *Track*, *Reacquire*) and the time elapsed since the last percept was anchored. If an anchor is grounded, the signature contains the data that is needed by the AF module to update the person attributes based on the *fusion model*. It is important to note that the anchor data from the individual anchoring processes is sent to the AF module asynchronously and no common time scale needs to be established between the component anchoring processes.

The person attributes that are updated with the signatures of the grounded anchors are the angle ϕ_{person} and distance d_{person} relative to the robot, the face height h_{person} and the person name. The initialization of the person attributes ϕ_{person} and d_{person} is carried out if the leg anchor is grounded for the first time. Then the *Find* function of the face anchoring process is started. The person is grounded if at least one of the component anchors is grounded. During normal operation the person's position is smoothly updated by the person's fusion model. Figure 6 shows the framework for anchoring the symbol *person*.

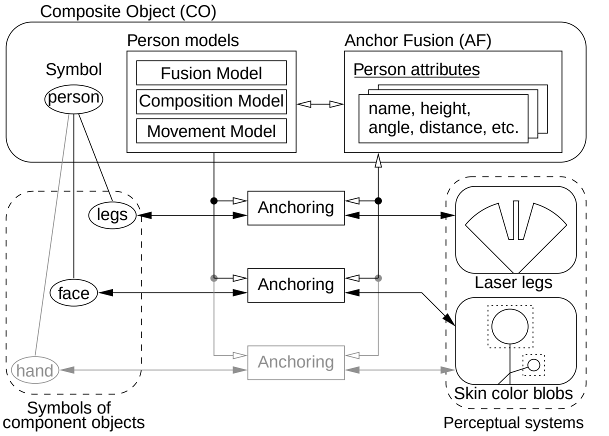

Fig. 6. Anchoring a person by anchoring the component symbols *legs* and *face*

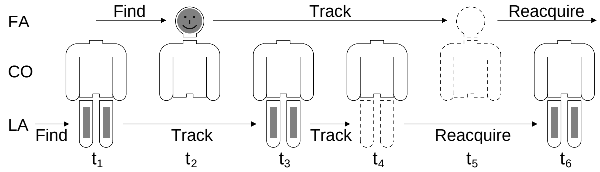

Fig. 7. A schematic example for anchoring a person.

To illustrate the concept a schematic example for anchoring a person is shown in Figure 7 depicting six consecutive timesteps at the beginning of an anchoring process:

t_1: Person anchoring is started and all component anchoring processes are in find mode. The leg detection generates a leg percept and the component symbol *legs* is anchored for the first time. Subsequently the person attributes in the AF module are initialized. Now an initial person position is available and the *Find* function of the *face* anchoring points the camera into the right direction.

t_2: The face detection generates a face percept and the component anchor for face is established. The FA switches from *Find* to *Track* and the person position in the AF module is updated with the grounded face anchor.

t_3: Again, the leg detection generates a percept of legs. Based on the *Track* function, the anchor for legs as well as the person attributes are updated.

t_4: In this time step, new laser range data is processed but no legs percept matching the LA is found. The anchoring process for legs switches from *Track* to *Reacquire*. No updating of the person attributes takes place.

t_5: A new camera image is processed but no face percept matching the prediction of the person position is found. The face anchoring process switches from *Track* to *Reacquire*. Now the person is ungrounded since neither the *legs* symbol nor the *face* symbol is grounded.

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

VI. Implementational Results

The anchoring of the component symbols is implemented in an object-oriented manner using C++. The individual anchors are derived from the basic anchor class and percept-specific data structures are added. The generic anchoring functions *Find*, *Track*, and *Reacquire* are defined in the basic anchor class while the functions for prediction, verification, selection and updating are defined by overloading specific implementations in the derived anchor classes.

We added our person tracking to the ISR software on the behavior level. When the robot is instructed to track a person the tracking behavior is started in parallel with other behaviors necessary for, e.g., obstacle avoidance. The tracking behavior initializes the AF process to anchor the person which in turn initializes all anchoring processes for the component symbols. The component anchoring processes retrieve percepts from the perceptual algorithms and send the anchor data to the AF module which sends the updated person's position to the tracking behavior that controls the robots motion. Ongoing work aims at using the person name and height for realizing an attentive HRI.

For a typical example of a person tracking scenario the state of the *person* anchor as well as the anchors for the component symbols together with some percepts are depicted in Fig. 8. Currently, the laser scanner provides 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 is negligible. The adaptive skin-color segmentation currently processes images with a size of 189×139 pixels. For each skin-colored region the face detection is carried out. The processing time of the overall face detection system depends on the number of skin-colored regions present in the image. For an image with two skin-colored regions the image processing running on the onboard PC provides percepts at a rate of around 3–4 Hz. Again, the time necessary for anchoring the percepts in the face anchor and combining the component anchors in the AF module is negligible. The person attributes are typically updated with a rate of 5–6 Hz due to the asynchronous anchoring of the different types of percepts leading to a partially asynchronous updating of the AF module.

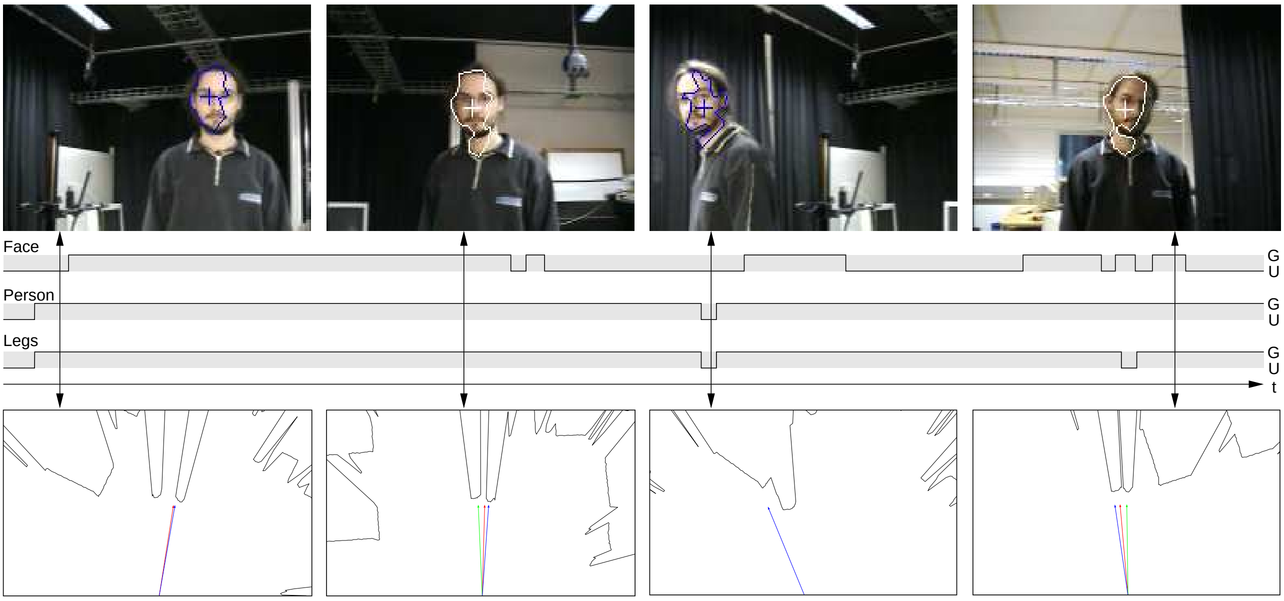

Fig. 8. The top row shows camera images with the polygons of skin-colored regions. A white polygon indicates a successful face detection. Below, the anchoring status of the symbols is shown. At the bottom the laserscans corresponding to the images are depicted. For a movie of this example see [1]

VII. Summary

We have presented a method for anchoring composite symbols through anchoring the component symbols with their associated percepts and subsequently fusing the resulting data of the component anchors. This modular approach facilitates distributed and multi-modal anchoring of component symbols 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*. The anchor fusion framework allows for multi-modal tracking of the person and integration of the different information cues to obtain an improved tracking performance. Through taking advantage of the different sensor capabilities in terms of precision and information content a more complete representation of the person to be tracked is maintained.

References

[1] http://www.techfak.uni-bielefeld.de/ags/ai/projects/mobile/.

[2] A. Agah. Human interactions with intelligent systems: research taxonomy. *Computers & Electrical Engineering*, 27(1):71–107, November 2000.

[3] H.-J. Böhme, U.-D. Braumann, A. Brakensiek, A. Corradini, M. Krabbes, and H.-M. Gross. User localisation for visually-based human-machine-interaction. In *Proc. IEEE Int. Conf. on Automatic Face & Gesture Recognition*, pages 486–491, 1998.

[4] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th AAAI Conf.*, pages 129–135, 2000.

[5] S. Coradeschi and A. Saffiotti. Perceptual anchoring of symbols for action. In *Proc. of the 17th IJCAI Conf.*, pages 407–412, 2001.

[6] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[7] St. Feyrer and A. Zell. Robust real-time pursuit of persons with a mobile robot using multisensor fusion. In *6th Int. Conf. on Intelligent Autonomous Systems (IAS-6)*, pages 710–715, Venice, 2000.

[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 *IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN)*, 2002. to appear.

[9] A. Gern, U. Franke, and P. Levi. Robust vehicle tracking fusing radar and vision. In *Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 323–328, 2001.

[10] M. Lindstrom M. Andersson, A. Oreback and H.I. Christensen. Intelligent sensor based robotics. Ch. ISR: An intelligent service robot, 1999.

[11] Jamie Sherrah and Shaogang Gong. Fusion of perceptual cues for robust tracking of head pose and position. In *Pattern Recognition, special issue on Data and Information Fusion in Image Processing and Computer Vision*, 2000. in press.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.

[13] J. Vermaak, A. Blake, M. Gangnet, and P. Perez. Sequential monte carlo fusion of sound and vision for speaker tracking. In *Proc. International Conference on Computer Vision*, volume 1, pages 741–746, 2001.

[14] S. Waldherr, S. Thrun, and R. Romero. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.