Real-time dynamic visual gesture recognition in human-robot interaction

Florian A. Bertsch and Verena V. Hafner

Abstract— **This paper tackles a common problem in human-robot interaction: recognizing the intentions of a human in an intuitive way. We present a system that is able to recognize dynamic human gestures in an interaction scenario where a humanoid robot visually observes the behavior of a human. This allows for a natural human-robot communication where no markers or technical devices are necessary on the side of the human interactor. The system not only recognizes previously learned gestures, but is also able to categorize and learn new gestures in an unsupervised manner. The proposed approach stands out due to its low computational cost and therefore can be used with the potentially slow embedded hardware of a humanoid robot. To demonstrate the possibilities of the approach we arranged a human - humanoid interaction game which consists of an alternating gesture-based communication.**

I. INTRODUCTION

Human-robot interaction is one of the main challenges and prerequisite for many applications in the field of robotics today. The more robots become a part of our everyday life the more it becomes crucial to develop simple and natural ways to interact with them in a way comparable to the interaction between humans. Human-humanoid interaction should therefore focus on intuitive ways of interaction such as natural gestures and the robot should be able to sense these gestures without any additional markers or technical equipment on the side of the human interactor. Due to the similarity of humanoid robots in looks, shape and morphology to humans the humanoid is not only able to recognize human gestures, but also to perform gestures by itself. This allows for a gesture-based interaction between human and humanoid as we will demonstrate with an interaction game.

To provide the ability of gesture recognition without additional equipment the method has to rely on visual observations captured by a video camera. A lot of work had been done in the field of video-based human motion analysis which led to many different approaches. Many of these approaches are not feasible for implementation on a humanoid due to their high computational costs. This is in particular the case for the model-based approaches that try to estimate a relationship between the observed image and a 3-dimensional model of the human body [10].

We therefore focus on gestures that can be described by the hand's movement within the image plane. This restriction avoids the problem of reconstructing the 3-dimensional human posture from an image and leads to a fast and accurate gesture recognition method.

Cognitive Robotics Group, Department of Computer Science, Humboldt-University Berlin, Germany
{bertsch,hafner}@informatik.hu-berlin.de

Promising methods in terms of fast calculation are color-based approaches that model the human skin-color to identify skin colored areas within the video frames to locate the human's head and hands. Color-based object localization and tracking has been successfully applied to many computer vision tasks but has the drawback of being prone to changes in illumination conditions [9]. To avoid that a manual color calibration is necessary, we introduced a first localization step, where a human who entered the humanoid's view was located by the typical shade produced by a human face. The colors within the area covered by the detected face were used to initially calibrate the model of the skin-color under the current illumination conditions.

The most common approach for the gesture recognition task is based on Hidden Markov Models (HMM). Since they proved to be excellent in their performance of handling time series with time varying characteristics in the field of speech recognition, they have been adopted to many problems including gesture recognition. Typically they are applied to time series of features that describe the gestures like the trajectories of involved body parts [13].

In addition to the HMM-based approach we introduced two simple approaches which use histograms and a decomposition of the trajectories into basic motions as basis for the recognition. These simple approaches turned out to be as good as the HMM-based approach in the case of our sample gesture set and could be calculated much more efficiently.

In the entire field of robotics learning plays an important role. Robots should be able to adapt their behavior to their environment and the needs of their users. This requirement is met by the described approaches since they use supervised learning techniques to learn new gestures based on a sample data set. In addition, clustering methods for unsupervised learning of new gestures were introduced which allowed another kind of learning. They group unknown gestures into clusters of gesture types due to their similarity without any additional information. This should be understood as an attempt to develop methods that enable a humanoid to learn new behaviors by observing a human who not necessarily pays attention to the humanoid.

II. VIDEO-BASED HUMAN MOTION TRACKING

In this section, we will describe the setup and methods for recognizing and tracking persons from a video stream.

A. Methods for recognizing and tracking human gestures

Many gestures can be described by the motion patterns generated by the upper body. The spatial movement of the

hands, in particular, contain meaningful information about a performed gesture. We therefore focused on the time series of the hands' movements as a basis for our gesture recognition system. It is known to be difficult to reconstruct the 3D configuration of the human body parts if a 2D image is the only data source [10]. In our approach, we avoided this reconstruction which typically comes along with high computational costs. Instead, we restricted our approach to gestures that can be described by the hands' movement within the image plane.

A commonly used approach to locate objects within a robot's view is to use the color as the discriminating feature, since it is fast and easy to compute but has, nevertheless, a high discriminative power and is robust against geometrical transformations [9]. Although most parts of the human body are covered by clothes of varying colors and textures the human face and the hands are typically bare. Hence it is possible to use skin-color as a feature to locate and track the face and the hands. By using color one must take, however, into account the importance of illumination conditions. A change of illumination can lead to a strong change in skin-color in the image.

A commonly used method to locate and track humans faces within video sequences is known as camshift [3]. We follow this approach by modeling the skin color in a way that adapts to varying illumination conditions and by using the resulting model to identify the center and shape of skin colored regions. There are two main differences to camshift: First we explicitly exclude colors that are not likely to be skin colors even under varying illumination. Secondly camshift has a good performance since it examines only a search window around the last object location. In our case we are interested in tracking multiple skin colored objects within the robots view. To avoid a mix-up of these objects and to ensure a useful tracking result in cases where the location of one object does not overlap between two consecutive frames (e.g. low frame rate) we use an approach that locates all skin colored objects within the robot's view and that finds a mapping between the actually located objects and the objects within the last frame in a second step.

The range of possible colors the skin could adopt is large. It is therefore not applicable as feature to locate head and hands without an appropriate calibration to the current illumination. We therefore used a different approach to locate a human who is entering the robot's view. This first localization is performed using a modification of the Viola-Jones face detector that relies on the typical pattern generated by a human face within a gray scale image [7]. This approach is considered too slow for real-time tracking of the face, but fast enough to recognize the presence of an appearing face within an acceptable delay. If a face is recognized, the colors within the area covered by the face are used to initialize a color model that describes the skin-color under the current illumination conditions (see Fig. 1).

The resulting color model can be improved by considering a pre-calculated set of colors which are not expected to be skin-colors even under changing illumination conditions.

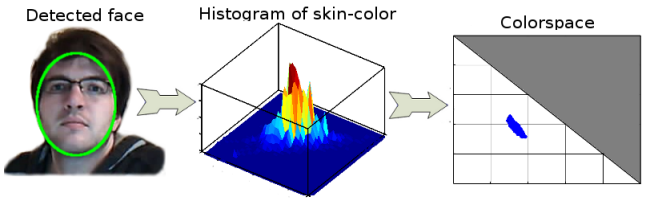

Fig. 1. The skin-colors within the area covered by a detected face (visualized as a green circle) form a compact cluster within the normalized RGB color space

Therefore, we pre-calculated the area of all possible skin-colors under a certain limited illumination variation which is called skin locus [9]. The shape of the skin locus is camera-dependent and can be determined by manually selecting skin-colored regions within a set of images taken under varying illumination cases. The skin locus is described within the normalized RGB color space, because it has been shown that the skin colors of different ethnicities overlap and form a small compact cluster [9]. The normalized RGB coordinates can be calculated as

$$nR = \frac{R}{R+G+B};\ nG = \frac{G}{R+G+B};\ nB = \frac{B}{R+G+B}$$

where R, G and B are the components of the RGB model. It is sufficient to use only the two components nR and nB due to the redundancy $nR + nG + nB = 1$.

A single Gaussian distribution is used for the color model that describes the skin-color during the current illumination conditions within the color space spanned by the normalized RGB components nR and nB. The model is initially fitted to the colors within the area of a detected face and then truncated to not exceed the skin locus. To handle changes within the illumination conditions during the tracking, the skin-color model is periodically adapted toward the colors within the area currently covered by the estimated face. This adaptation is performed by using exponential smoothing, whereas the skin locus is taken into account to ensure that the color model does not adapt to non-skin objects.

Using the color model we localized skin-colored, connected components within the image which are called blobs. To perform a suitable tracking of skin-colored objects within the scene, we hypothesized a relation between the currently observed blobs and a set of skin-colored objects we were tracking. This is done following an approach described in [1] which can handle multiple skin-colored objects that may overlap each other considering a possibly moving camera. The spatial distribution of the currently tracked object is described by an ellipse (see Fig. 2). In the original approach the distances between each point that belongs to a blob and the elliptic models of each tracked object are calculated. Since this pixel-wise distance calculation is too inefficient to be calculated on the embedded hardware of many humanoid robots, we modified the approach to reduce the distance calculations. We therefore calculated the oriented bounding boxes of all tracked objects and all observed blobs. Based on a collision detection of the oriented bounding boxes we

were able to reduce the pixel-wise distance calculation to the blobs which are covered by multiple objects.

Fig. 2. Currently tracked skin-colored objects described as ellipses.

Within the set of skin-colored objects the objects that represent the head and the hands of the observed human are determined. To simplify this step and to avoid the need of multiple color models associated with different people's skin we assume that only one person is within the robot's view.For each of the body parts we are selecting the object that is most similar to the object that described the body part during the last time step, where the similarity is considered in terms of spatial adjacency and shape similarity. The most similar object is regarded as the current observation of the body part and is used as input for a Kalman filter [6]. The initialization of the tracking of the head is done by using the initially detected face as first observation. A hand object is initialized by using the largest object that is visible for some time without changing its location too much. To avoid a mix-up of the head and the hands, some additional rules are used for differentiation based on common configurations of the body parts and the fact that the head is usually more stable regarding its position than the gesticulating hands. To give an impression of the tracking results Fig. 3 shows some sample frames of the tracking of a human's head and hands during a waving gesture.

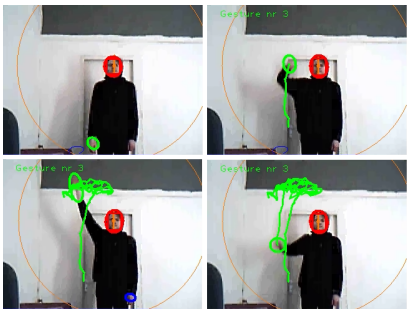

Fig. 3. Tracking of the head and the hands during a waving gesture

B. Feature selection

The tracking of a human head and hands within the image are used as basis for the gesture recognition. To extract features that describe the observed gestures in an appropriate way, we applied some additional preprocessing steps to the resulting location time series of the body parts. The first step is a normalization which is used to compensate different distances and spatial offsets of the gesticulating human within the robot's view. The head's position and size is used as reference for the normalization. The positions of the hands are transformed to coordinates that are relative to the head's position and scaled in proportion to the head's width. If a hand's position exceeds a circular area around the head it is truncated. The radius of this circular area reflects the typical proportion of a human's head width and arm's length. As a result we obtained a time series of positions within a circular area which is transformed to have an origin of $(0, 0)$ and a radius of 1.

The next preprocessing step is used to identify segments within the continuous sequence of normalized positions which describe a single gesture. To make this segmentation possible we assumed that the gesticulating person returns to a resting posture between performing gestures. The person remains in the resting posture if both arms hang down beside the torso. Therefore, a gesture is defined as any segment of the continuous sequence of normalized positions where at least one hand is not located within an area describing the possible resting positions.

After the normalization and segmentation each gesture is described by the time series of two hand positions. In a series of comparative experiments we found that the best way to encode this information for a further processing is to describe the position as well as the direction and velocity of movement for each time step by using the current position and motion vector. The positions and motion vectors of both hands are alternately inserted into a common feature sequence, whereby the artificial position $(-1, -1)$ and the motion vector $(-1, -1)$ are inserted if a hand is currently not tracked. Finally, if a method is used for further processing that cannot handle continuous values then the Linde-Buzo-Gray (LBG) method [8] for vector quantization is applied to transform the time series of continuous values to a time series of discrete symbols (see Fig. 4).

Fig. 4. Results of the Linde-Buzo-Gray vector quantization method used to split the normalized feature space into 16, 32 and 64 sectors that can be used to transform the continuous features into discrete symbols

C. Basic Motions

A higher abstraction level to represent the observed gestures than the described feature sequence can be achieved by a decomposition of the gestures into a set of basic motions. This decomposition is going along with a more compact representation of the gestures which turns out to be appropriate as basis for a reliable gesture recognition. A basic motion is defined as a part of one hand's movement without a significant change of the moving direction. To decompose one hand's movement into basic motions we determined the

positions with a significant change in movement and approximated the motion in-between with directed line segments, which we called the basic motions. Each basic motion is described by the position where it starts and the position where it ends. This leads to a sequence of directed line segments that describe the important movements of a hand while ignoring the unimportant variation of the movement in-between.

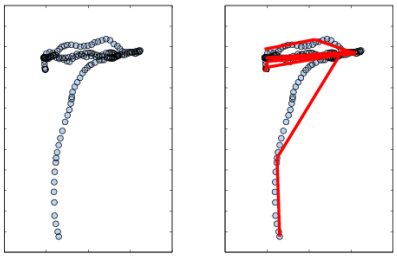

Fig. 5. Decomposition of the trajectory of a waving gesture into basic motion segments

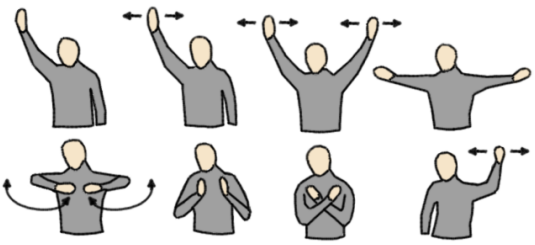

Fig. 6. Set of sample gestures

Fig. 7. Sample pictures from the video capturing of 9 persons performing 212 gestures of 8 different types

III. LEARNING OF GESTURES

In this section, we will describe the methods applied to the extracted features for learning and recognizing gestures.

A. Experimental setup and choice of gestures

To ensure that our method can be calculated in real-time on a humanoid robot we focused on gestures that can be described by the movement of a human's hands within the image plane. This condition restricts the gestures we may use to such cases where the gesticulating person is frontally oriented toward the humanoid and uses "large-scale" movements of the hands. While people typically use mimic and finger gestures of smaller size during a conversation, they use "large-scale" movements if they try to gesticulate over large spatial distances. Therefore, we choose eight sample gestures out of a set of gestures used by construction workers to instruct vehicle drivers (see figure 6). To avoid accidents they ensure an unmistakable communication by using gestures that fulfill the described conditions.

The type of gestures we are dealing with is distinct from the types considered in existing approaches used for visual gesture-based human-robot interaction. Most approaches use hand signs as gestures. Others, that use the tracking of skin-colored areas focus on signs that are "drawn" with one hand within the image area. The approach described in [5] handles more natural gestures that can be expressed by the human posture, as in our case, but is limited to static poses.

The 8 different gestures were performed by 9 different persons and recorded with a video camera. This resulted in a database of 212 gestures in total. The recorded persons wore normal clothes which covered the arms and legs so that the face and the hands were the only uncovered parts of the body. They were asked to perform the gestures slowly and make sure that the palms of their hands were always orientated toward the camera. Some sample images taken from the recorded videos are shown in Fig. 7 to give an impression of the conditions during the video capturing.

B. Supervised learning of a set of gestures

A widely-used set of methods for the gesture recognition task are Hidden Markov Models (HMM) [13]. They use a state machine of hidden states to represent time-varying changes of the modeled process. Each hidden state is linked with a distribution that describes the probability of observing a value or symbol while the hidden state is active. If the HMM changes its hidden state, this is going together with a change of HMM's output probabilities. If these probabilities are described by a single Gaussian distribution or by a mixture of many the HMMs are called "Mixtures of Gaussian Hidden Markov Models" (MHMM). The MHMMs are used to model time series of continuous values. In contrast, there is another class of HMMs that is used to model time series of discrete symbols. For this purpose they describe the probability of each of the symbols to be observed during each hidden state. This kind of model is called a "Discrete Hidden Markov Model" (DHMM). To be able to use DHMMs to recognize gestures we have to express the gestures as a sequence of discrete symbols. This can be done by using a vector quantization method that transforms a sequence of continuous features into a sequence of discrete symbols (e.g. the LBG method [8]).

During our experiments to determine the best number of hidden states for the DHMMs used to recognize the sample gestures, we frequently observed that we obtained the best results when only using one hidden state. This case reduces the complexity of a DHMM since there are no transitions between hidden states. Such a simple DHMM consists of only one discrete distribution that describes the observation probability of each possible symbol. It is obvious that such

a DHMM can be expressed in terms of a histogram with a bin for each symbol that describes the symbol's relative frequency and thus its estimated observation probability. An advantage of using a histogram is the avoidance of costly methods to train HMMs. We therefore considered the histogram-based gesture recognition as a fast alternative solution. This approach can be used to recognize gestures that can be described by a process without temporally changing characteristics - as it seems to be the case for our sample gestures.

C. Learning by observation

In addition to the approach described in the last section that allows to learn gestures in a supervised manner from a data set where each gesture type is known in advance we developed a method to learn unknown gestures by observation. Therefore, it is possible to perform a sequence of gestures not known to the humanoid and it has the ability to learn new gesture types by grouping the presented gestures. This allows the humanoid to learn observed gestures without any additional information. Moreover, it is an attempt to develop methods that enable a humanoid to learn new behaviors by observing a human who not necessarily pays attention to the humanoid. In this sense it is a modification of the famous "programming by demonstration" [4] concept toward a "learning by observation" concept.

D. Clustering Methods

Since the "learning by observation" concept in our case is nothing else but a typical unsupervised learning approach, we compared the hierarchical clustering and the k-means clustering applied to our data set of sample gestures. To measure the agreement between the resulting clusters and the real division into gesture types we used the adjusted Rand index [11], a measure for similarity often used in clustering. The possible resulting values of the adjusted Rand index range from 0 to 1, whereby large values stand for a great similarity between the clusters and the real division into gestures.

Our first approach to describe the gesture similarities is to encode the gestures as vectors of the same length. Using such an encoding we could apply common metrics to calculate the distance between the gestures, hence their similarity. To encode the gestures as vectors we fitted one of the models which we described in the previous subsection (HMM or histogram) for each single gesture. We then used all parameters of each model to build a vector representing the corresponding gesture. We used a principle components analysis (PCA) to reduce the number of vector components as much as possible by keeping simultaneously at least 95% of the original data's variance. The city-block-metric-based distances between the resulting vectors were used to calculate the distances between the corresponding gestures.

As another approach to calculate the similarity between the gestures we defined a distance measurement based on the representation of the gestures as basic motions. Keeping the observation in mind that the gestures of our sample set could be recognized by histogram-based approaches without modeling time-varying dynamics, we did not consider any order within the basic motions. Instead, we defined the distance between two gestures as the mean of the minimum distances of every basic motion to all basic motions of the other gesture. To build this distance we took into account the pairwise most similar basic motions of each one of the compared gestures.

Based on our findings that a hierarchical cluster analysis using the basic-motion-based distance measurement leads to the best results (as will be described in section IV) we developed an online clustering method that can be used for our initially proposed "learning by observation" scenario. We used the combination of a hierarchical cluster analysis and the basic-motion-based distance measurement as core for a procedure to identify gesture types within a sequence of unknown gestures. Therefore, the procedure calculates clusters within the gestures whenever a new gesture was observed. The calculation of the clusters consists of the following steps:

1) The new gesture is assigned to the cluster whose gestures are most similar on average.
2) If a hierarchical cluster analysis can be used to form two clearly distinct sub-clusters within the cluster the new gesture has been assigned to, these sub-clusters are used to build two new gesture types.
3) From each gesture type the gesture that has the smallest average distance to all other gestures of the same type is chosen to represent the cluster. Each gesture that was not chosen to represent a cluster is assigned to the cluster whose representing gesture is the most similar.
4) If the average distance between the gestures of two types becomes clearly smaller than the average distances between the other types then the two types are merged together.
5) If a gesture type contains a very small amount of gestures (e.g. less than 3) it is deleted and its gestures are assigned each to that cluster whose gestures are most similar on average.

IV. RESULTS

In this section, we will first describe the accuracy we achieved using the described methods to recognize the set of sample gestures. Then we will present a simple interaction game between a human and a humanoid robot as application of the proposed method.

A. Accuracy of the gesture recognition

1) Recognizing a fixed gesture set: We used the HMMs and histograms to prove their performance in the gesture recognition task regarding our sample gesture set. This was done in a typical supervised manner. We used the data set of 212 gestures performed by 9 different persons covering all 8 gesture types as described in the experimental setup section. Based on this data set we compared different recognition methods using cross-validation in a "leave-one-person-out" manner. In the case of the HMM-based modeling as well

as in the case of a histogram-based modeling we used the training data to calculate a model for each type of gesture within the sample gesture set. These models were then used to recognize the 9th person's gestures by assigning each to that gesture type whose model describes it in the best way. The best results we achieved for each model type are given by a recognition rate of approximately 0.7 for DHMM, 0.9 for MHMM and 0.9 for histogram-based methods.In the case of the DHMM we used 4 hidden states and the vector quantization into 256 symbols. The MHMM used 4 hidden states, too, and the histogram-based approach used the same vector quantization as in the case of the DHMM. Our result shows a comparable performance for the MHMM and the histogram-based approach, namely a recognition rate of approx. 0.9 (in comparison to $1/8$ for a random guess). This result can be expected to become better in the case when the gesticulating person gains experience with the gesture recognition system.

2) Learning unknown gestures by observation: To evaluate the quality of the different cluster methods we applied them to the 212 gestures of the sample data set. To that end, the similarity between these gestures was represented by the different approaches described above to determine an appropriate combination of clustering method and gesture representation. The clustering was aimed at achieving 8 clusters which correspond to the number of gesture-types within the data set. This pre-defined number of gesture types is normally not known, but used in this case to compare the results.

The results of the comparison between the cluster analysis methods applied to the different representations of the gesture's similarities showed the best results when using the parameters of DHMMs trained for each gesture in the case of representing each gesture as a vector. In this case the k-means method which produced a mean adjusted Rand index of approx. 0.75 was slightly better than the hierarchical method that produced an adjusted Rand index of approx. 0.70. In the case of the basic-motion-based distance measurement the hierarchical method was the better one producing an adjusted Rand index of approx. 0.8, whereas the k-means method produced a mean adjusted Rand index of approx. 0.65. Comparing the methods, the hierarchical cluster analysis might be preferred, because it produced similar or better results without a need to restart the calculation several times with different random initializations as it is the case with the k-means method. When comparing the representation of the gesture similarities we found that the basic-motion-based distance measurement outperformed the vector-based approach and had the additional advantage that its calculation is noticeably faster than the vector-based approach because it does not need to fit a DHMM for each single gesture.

To evaluate the online cluster method we generated random sequences from the data set of 212 gestures. For each random sequence we randomly chose 5 gestures of each gesture type and inserted them in random order into the sequence. Using the generated random sequences we performed a grid-search-based optimization of the parameters that controled the details of the online clustering procedure. As result we obtained a parameter set that led to an adjusted Rand index of 0.70 on average by clustering 50 random sequences. To give an idea of this result we visualized the result for 8 randomly chosen random sequences in Fig. 8. Each disk of this visualization represents one gesture whereas the spatial distribution into groups shows the result of the online clustering procedure, and the color of each disk represents its gesture type. The visualized results of the 8 chosen sequences are varying regarding their quality. The true number of gesture types was achieved in 4 cases. The other cases are differing by one type from the true number (3 x 7 and 1 x 9 types). The quality of the assignment from the gestures to the types is varying as well. The best result is shown in the lower left corner where the 8 resulting clusters are corresponding unambiguously to the gesture types and only 3 gestures out of the 40 are assigned to the wrong type. The other tests show a less clear clustering but the gestures are far from being randomly distributed.

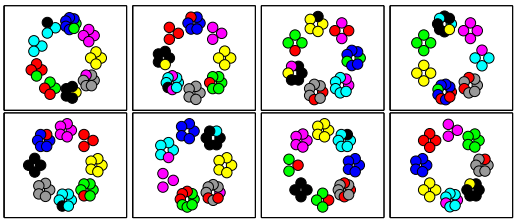

Fig. 8. Visualization of the unsupervised online learning result of eight random sequences of gestures

B. Application to a human-robot interaction scenario

The method for extracting and learning human gestures from video streams described above has already been optimized for implementation on a humanoid robot and is tested in the interaction game scenario described in this section.

1) Humanoid robot: The humanoid robot used in these experiments is a Nao [14] from Aldebaran Robotics with the following hard- and software:

- AMD GEODE 500 MHz processor, 256 MB SDRAM
- Embedded Linux (32bit x86)
- video camera (640x480 resolution at 30 fps)
- Universal Real-time Behavior Interface (URBI) [2]

To allow for a real-time gesture recognition we improved the runtime of the proposed method by downsampling the image resolution to 160x120 and changing the image representation to a compact run length encoding by the first processing step.

2) Interaction game: To give an impression of possible applications of the proposed gesture recognition method, we arranged a gesture-based interaction game. To demonstrate the humanoid's gesture recognition skill as well as its ability to use its human-like shape to perform gestures by itself, the game consists of alternating gesture recognition and presentation tasks for both participants. The humanoid

opens the game by presenting one gesture of the 8 sample gesture types, which should be repeated by the human. The humanoid then uses its speech synthesis capability to give a feedback if the recognition result of the observed answer gesture matches the gesture type that was initially performed by the humanoid. Now the participants change their roles and the human presents a gesture which is answered by the humanoid that is performing the gesture type which it has recognized. These role changes can be repeated in alternation.

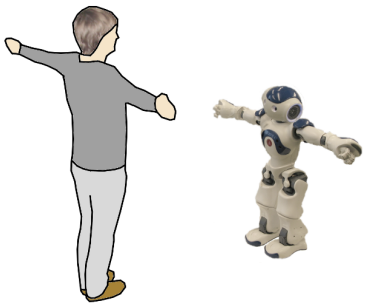

Fig. 9. An interaction game demonstrates a gesture-based mutual human-humanoid interaction

V. CONCLUSIONS AND FUTURE WORK

A. Conclusions

By using the proposed approach it is possible to perform a real-time visual recognition of dynamic human gestures with high accuracy. Since there is no need of additional devices it provides a simple and natural kind of human-humanoid communication. To achieve an accurate real-time recognition based on the potentially slow embedded hardware of a humanoid we focused on gestures that can be described by the hands' motions within the image plane. This is a strong restriction but it allows to avoid the hard reconstruction task of 3-dimensional body configurations using only an image as information source. The used sample gesture set shows, that there are simple and natural gestures that fulfill the constraint and therefore can be used as basis for a suitable interaction.

The results of the comparison of different methods for supervised gesture learning and recognition showed similar results for the HMM- and the histogram-based approaches. It is striking that the histogram-based approach, which is much simpler than the HMM approach, results in a comparable performance in the case of our sample gesture set. This allows to avoid the non-linear optimization problem to train HMMs and leads to an efficient training procedure. These findings led to a similar simple solution for the proposed unsupervised "learning by observation" task. By calculating the distances of gestures based on their decomposition into basic motions without considering their timely order, we achieved a powerful and simple solution for the clustering task. The online clustering method that is based on the simple distance measurement of gestures showed promising results.

The implementation on the humanoid robot Nao demonstrated that the proposed method could successfully be used to equip humanoids with gesture recognition skills. We used these skills to give an impression of the possibilities of a gesture-based human-humanoid interaction by arranging an appropriate interaction game.

B. Future Works

The presented approach of learning by observation allows learning the recognition of unknown gestures without an explicit training session. Therefore it provides a possibility to increase the set of known gestures continuously. It would be desirable to extend this skill in a way that the humanoid is not only able to recognize the new gestures but also to perform them in addition to the predefined set of gestures. This would be a typical human-robot imitation task [12] which seems to be easy to achieve when using the gestures we focused on. Since the gestures are described by the hands' movement parallel to the human's orientation we can easily set up a relation between the observed hand positions and the humanoid's posture. Using such an imitation skill the humanoid would be able to give visual feedback of an observed gesture which could be used to verify that the gesture was observed correctly.

REFERENCES

[1] A. A. Argyros and M. I. A. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera", *ECCV*, 2004, pp. 368–379

[2] J. C. Baillie, "Design Principles for a Universal Robotic Software Platform and Application to URBI", *IEEE ICRA 2007 Workshop on Software Development and Integration in Robotics*, 2007

[3] G. R. Bradski, "Real Time Face and Object Tracking as a Component of a Perceptual User Interface", *Fourth IEEE Workshop on Applications of Computer Vision (WACV'98)*, 1998

[4] Y. Demiris and A. Billard, "Special Issue on Robot Learning by Observation, Demonstration, and Imitation", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 37, Issue 2, 2007, 254–255

[5] Md. Hasanuzzamana, T. Zhanga, V. Ampornaramvetha, H. Gotodaa, Y. Shiraib and H. Uenoa, "Adaptive visual gesture recognition for humanrobot interaction using a knowledge-based software platform ", *Robotics and Autonomous Systems*, vol. 55, Issue 8, 2007, 643–657

[6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, 1960, pp. 35–45

[7] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", *IEEE ICIP*, vol. 1, 2002, pp. 900-903

[8] Y. Linde, A. Buzo and R. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, vol. 28, 1980, pp. 84–94

[9] B. Martinkauppi, M. Soriano and M. Pietikäinen, "Comparison of skin color detection and tracking methods under varying illumination", *Journal of Electronic Imaging*, vol. 14, 2005

[10] R. Poppe, "Vision-based human motion analysis: An overview", *Computer Vision and Image Understanding*, vol. 108, 2007, pp. 4-18

[11] W. M. Rand, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association*, vol. 66, 1971, pp. 846–850

[12] J. Saunders, C. L. Nehaniv, K. Dautenhahn and A. Alissandrakis, "Self-Imitation and Environmental Scaffolding for Robot Teaching", *International Journal of Advanced Robotics Systems, Special Issue on Human - Robot Interaction*, Vol. 4, Issue 1, 2007, pp. 109–124

[13] J. Yang and Y. Xu, "Hidden Markov Model for Gesture Recognition," tech. report CMU-RI-TR-94-10, Robotics Institute, Carnegie Mellon University, 1994

[14] NAO Humanoid Robot. Aldebaran Robotics, Paris, France. *http://www.aldebaran-robotics.com/eng/* Date Last Accessed: June 30th, 2009.