Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

International Journal on Artificial Intelligence Tools
Vol. 0, No. 0 (1999) 000—000
© World Scientific Publishing Company

(Leave $1\frac{1}{2}$ inch blank space for Publisher)

A HYBRID STOCHASTIC-CONNECTIONIST APPROACH TO GESTURE RECOGNITION

ANDREA CORRADINI, HANS-JOACHIM BOEHME, and HORST-MICHAEL GROSS
Ilmenau Technical University, Department of Neuroinformatics
P.O.B. 100565, 98684 Ilmenau, Germany

Received (received date)
Revised (revised date)

In this paper a person-specific saliency system and subsequently two architectures for the recognition of dynamic gestures are described. The systems implemented are designed to take a sequence of images and to assign it to one of a number of discrete classes where each of them corresponds to a gesture from a predefined small vocabulary. Since we think that for a human-computer interaction the localization of the user is essential for any further step regarding the recognition and the interpretation of gestures, in the first part, we begin with describing our saliency system dedicated to the person localization task in cluttered environments. Successively, the intrinsic gesture recognition process is broken down into an initial preprocessing stage followed by a mapping from the preprocessed input variables to an output variable representing the class label. Subsequently, we utilize two different classifiers for mapping the ordered sequence of feature vectors to one gesture category. The first classifier utilizes a hybrid combination of Kohonen Self-Organizing Map (SOM) and Discrete Hidden Markov Models (DHMM). As second recognizer a system of Continuous Hidden Markov Models (CHMM) is used. Preliminary experiments with our baseline systems are demonstrated.

Keywords: person localization, gesture recognition, hybrid stochastic-connectionist system, Hidden Markov Model, self organizing map.

1. Introduction

Gestures are part of everyday natural human communication. They are used as an accompaniment to spoken language and as an expressive medium in their own right. Recently, there have been strong efforts to develop intelligent, natural interfaces between users and systems based on gesture recognition. The optimal interaction has to be natural, intuitive, not require any remembrance and is similar to that we are familiar, thus the interaction with other people. Such intelligent interfaces cover a broad range of application fields in which an arbitrary system is to be controlled by an external user or in which system and user have to interact immediately [1,2,3]. In our special case, we aim at a naturally behaving human-robot interface that

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

combines different, especially visual and auditory sensor modalities. This interface is to be used as a framework for intelligent human-robot interaction in service-system domains. In this paper, we exclusively concentrate on the description of the visual part of that interface.

We state that a proper person localization is an absolute prerequisite for any further gesture recognition process, especially in cluttered and un-engineered environments. Therefore, we propose a person-specific saliency system combining different feature modules into a multiple-cue approach. The features at hand are *skin color*, *facial structure*, and *structure of the head-shoulder-contour* respectively. The utility of the different parallel processing cue modules is to make the saliency system robust and independent of the presence of one certain information source in the images. Hence, we can handle varying environmental circumstances much easier, which, for instance, make the skin color detection difficult or almost impossible. Due to its reliability and robustness against varying environmental conditions, this system represents the starting point for any further precessing step.

One of the crucial problems in recognition of gestures is the handling of their varying temporal and spatial structure. That difficulty stems from the high variability of each movement associated with a gesture to be detected. Gesture's segments may overlap, have varying lengths, and vary across speakers. Even the same user is not ever able to produce exactly the same movement for the same gesture. Moreover, the complexity of the automatic recognition task is related to robustness to environmental conditions, vocabulary size, number and movement characteristics of users in user independent recognizers, and so on.

This paper is structured as follows. Starting from our saliency system for person localization [4] (section 2), in section 3 we provide an overview of the process which is to be carried out to describe the user's postures. We propose to combine skin color-based image segmentation with shape analysis by means of invariant moments. Section 4 mentions some basic ideas of the theory of both Self-Organizing Maps and Discrete Hidden Markov Models, and section 5 describes how we exploit these tools for gesture recognition. An alternative stochastic architecture relied on Continuous Hidden Markov Models is suggested in Section 6. Finally, a description of the preliminary results and some final considerations can be found in Section 7 and Section 8, respectively.

Figure 1 introduces our mobile robot PERSES, a standard B21-platform (Real World Interfaces Inc.). This robot acts as the experimental system for human-robot interaction. Besides ultrasonic and infrared distance sensors PERSES comes with a multi-camera system, consisting of two frontally aligned cameras (stereo head) and one omnidirectional camera covering the surroundings of the robot by a circular image.

2. Saliency System for Person Localization

The saliency system for person localization is divided into two subsystems: one

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

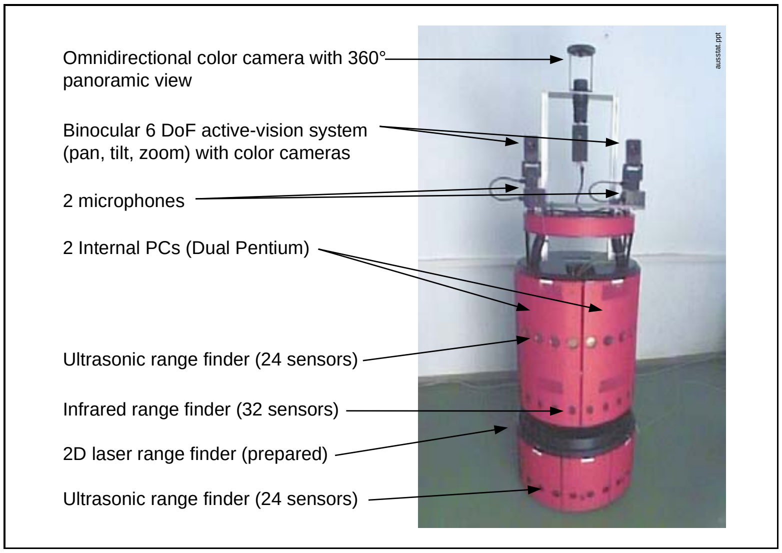
Omnidirectional color camera with 360° panoramic view
Binocular 6 DoF active-vision system (pan, tilt, zoom) with color cameras
2 microphones
2 Internal PCs (Dual Pentium)
Ultrasonic range finder (24 sensors)
Infrared range finder (32 sensors)
2D laser range finder (prepared)
Ultrasonic range finder (24 sensors)

Fig. 1. The mobile robot PERSES.

subsystem uses the images coming from the omnidirectional camera and performs a motion-based foreground-background segmentation, whereas the second one processes the images acquired by one of the two frontally aligned cameras and applies a multiple-cue approach. The results of both subsystems are properly combined the yield a satisfying person localization. The motion-based segmentation gives us some candidate regions that indicate where persons could be in the surroundings of the robot. By turning the robot towards those candidate regions the multiple-cue approach analyzes that regions in much more detail and decides finally if there is a person that wants to interact with the robot.

2.1. *Motion-based segmentation of the omnidirectional images*

An omnidirectional view covers the surroundings of the robot in a one-shot manner, without the need of any movement (camera or robot). On the other hand, because of the rather low detail resolution of those images, a movement-based method seems to be a proper choice to detect moving objects (persons). Our implemented method is similar to that suggested in the *Pfinder* system [5], but differs in the following aspects: (i) The statistical models for foreground and background pixels were simplified to boxes, and (ii) the foreground and background models are continuously adjusted. The model simplification led to a lower computational complexity resulting in a performance speed-up, surprisingly without almost non lost in sensitivity. The latter is to take into account that the robot cruises its surroundings which makes it impossible to use only one stationary background model.

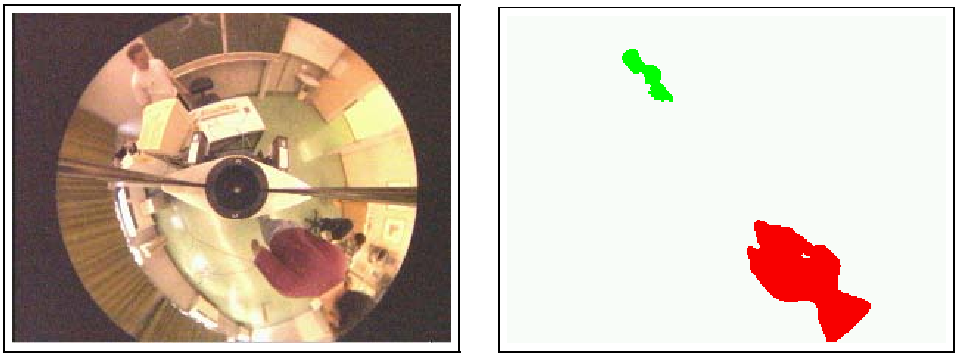

Fig. 2. Movement-based person detection. Left: original image. Right: segmented image; the two marked regions correspond with two persons at different distances to the robot.

After the alignment of all image pixels to the foreground and background model, respectively, a simple grouping mechanism is applied to get closed moving regions and to suppress noise and very little regions. Figure 2 illustrates the method.

The segmented candidate regions are labeled according to their distance to the robot. Subsequently, the robot will turn itself towards the nearest hypothesized person and apply a multiple-cue approach for person localization, which is described in the following subsection.

2.2. *Multiple-cue approach*

Figure 3 provides a coarse sketch of the multiple-cue approach for user localization. A multiresolution pyramid transforms the images acquired by one of the frontally aligned cameras into a multiscale representation. Two cue modules sensitive to *facial structure* and *structure of a head-shoulder contour*, respectively, operate at all levels of a grayscale pyramid. The cue module for *skin color* detection uses the original color image. Its segmentation result is transformed into a pyramid representation, too, to obtain an uniform data structure for the different cues. The utility of the different parallel processing cue modules is to make the saliency system robust and independent of the presence of one certain information source in the images. Hence, we can handle varying environmental circumstances much easier, which, for instance, make the skin color detection difficult or almost impossible. Furthermore, high expense for the development of the cue modules can be avoided (see [6,7], too).

The output of the cue modules serves as the input for the saliency pyramid at each resolutional level. The maps are topographically organized neural fields containing dynamic neurons interacting among each other (see [8,9]). In the saliency maps *all those regions* shall become prominent that most likely cover *the upper part of a person*.

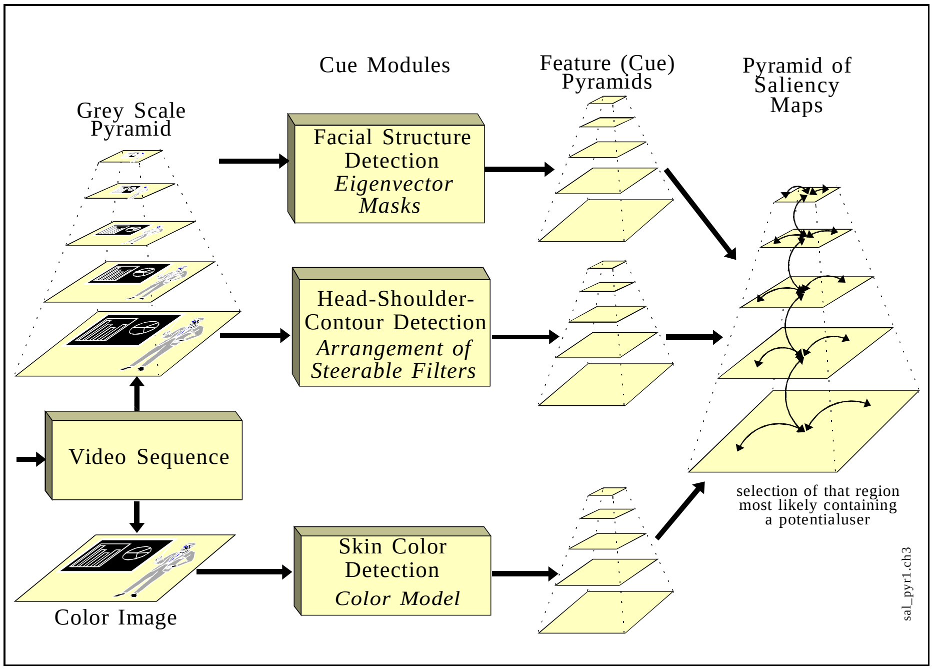

Fig. 3. Architecture of the multiple-cue approach for person localization.

In our previous work (see [10]) the three cues were assumed to be of equal importance. After a period of practical experiences, we had to face that the shape-based approach provides much more reliable contributions to the localization process compared to the skin color and facial structure cues. The reasons are quite obvious: Skin color detection is highly influenced by illumination. Although we use an additional color adaptation method (see [11]) to yield constant color sensation, robust skin color detection cannot ensured in general. Further, solving the localization problem becomes more interesting the farther away the person is. Necessarily, relevant features should appear even on rather coarse resolutional scales so that details, as facial structures, are less prominent. Facial structure can be detected confidently only if the distance between person and camera is not too large. Otherwise, the region covered by the face becomes to small to be localized.

Against this background, the method for head-shoulder-contour detection was improved significantly. The actual method is described in more detail in the following subsection. Since the other cues can only support the person localization, but cannot ensure the localization alone, their methods were reduced to rather simple, but computationally efficient algorithms. The following subsections describe the cues for person specific saliency in more detail.

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

2.2.1. *Head-shoulder contour*

The contour which we refer to is that of the upper body of frontally aligned persons. Our simple contour shape prototype model consists of an arrangement of oriented filters doing a piecewise approximation of the upper shape (head, shoulder) of a frontally aligned person. The arrangement itself was learned based on a set of training images. Applying such a filter arrangement in a multi-resolutional manner, this leads to a robust localization of frontally aligned persons even in depth.

Arrangements of steerable filters – motivation and related work: The idea of this method refers just to a description of the outer shape of head and shoulders and is based both on some physiological considerations as well as on psychophysical effects.

The visual cortex consists in several parts of cells with oriented receptive fields. A lot of investigations have shown that the profile of receptive fields of simple cells in the mammalian primary visual cortex can be modeled by some two-dimensional mathematical functions. Gaborian [12] and Gaussian functions (incl. low order derivatives) [13] appear to provide the typical profiles for visual receptive fields. So, local operations decompose the visual information with respect to the frequency space.

Psychophysical aspects for the contour-shape based approach, e. g., good continuation or symmetry (both belonging to the Gestalt laws), obviously describe effects which necessitate grouping mechanisms. Against this background, we have chosen the approach of an *arrangement* of oriented filters.

Because each section of the contour should be approximated by a special oriented filter, localizing a person would require possibly as many *differently oriented* filters as orientations belong to the arrangement. Since that would be computationally very costly, we turned to steerable filters.

Determining the course of contour: Steerable filters have the nice property that an a-priori limited number of convolutions is sufficient to derive any orientation information within an image. Thus, their use provides an extended set of orientations, avoids the necessity of numerous additional filters, and enables a more accurate computation of the course of contour.

Our complete data set consists of images showing ten persons in front of a homogeneous background under three different viewing angles ($0°, +10°$ and $-10°$, where $0°$ corresponds to an exactly frontally aligned body). All these images have been recorded under identic conditions (position, illumination, distance). Additionally, in order to achieve a symmetrical contour model the whole data set was vertically mirrored extending the data set to 60 images. Subsequently, the 256×256-images (grayscale) were low-pass filtered and scaled down to 16×16. Then, we applied a Sobel operator to the images enhancing the edges of each image. Next, all of those edge-marked intermediate images were averaged, since the contour to be determined *on average* should match the real outer contour. After this, we thresholded to find

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

that edge representing the typical contour shape.

This way, we got the course of the contour of interest resulting in a 16×16 binary matrix where the elements along the contour are set to 1, the others remain 0. We refer to this contour matrix, our template, as $\mathbf{\Lambda}^\star$. The local orientation of each contour element is determined by means of the steerable filters (see below). These are applied to the binary contour shape so that for each element of $\mathbf{\Lambda}^\star$ with value 1 an angle of orientation can be determined resulting in a matrix $\mathbf{\Lambda}$ (see figure 4).

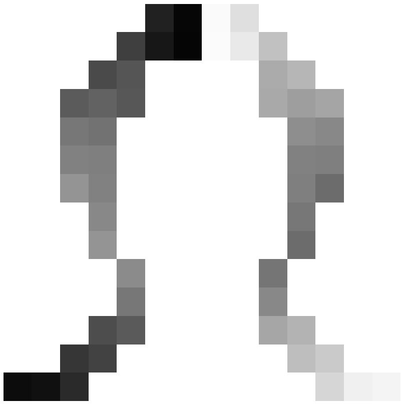

Fig. 4. The determined shape of contour $\mathbf{\Lambda}$: orientation angles coded by gray values (0°: black; 90°: medium gray; 180°: white). Note that around the forehead transitions from 180° to 0° occur. The contour shape is symmetric since the original data set was mirrored.

Applying steerable filters After determining the binary contour, we measure the local orientation by means of a set of filters which are oriented in every direction. We take the powerful approach of *steerable filters* (see [14]) for orientation estimation. It provides an efficient filtering output by applying a few *basis filters* corresponding to a few angles and then interpolating the basis filter responses in the desired direction. Steerable filters are computationally efficient and do not suffer from the orientation selection problem.

In general, a function f is considered to be steerable if the following two conditions are satisfied. First, its basis filter set is made up of M rotated copies of the functions $f^{\alpha_1} \dots f^{\alpha_M}$ on any certain angles $\alpha_1 \dots \alpha_M$. Second, a rotated copy f^ϑ of it on some angle ϑ has to be obtained by a superposition of its basis set multiplied by the interpolation functions $k_j(\vartheta)$ as in

$$f^\vartheta = \sum_{j=1}^{M} k_j(\vartheta) f^{\alpha_j} \tag{1}$$

In our work, we take a quadrature pair by using the second derivative of a Gaussian and an approximation of its Hilbert transform by a third-order polynomial modulating a Gaussian. From the steering theorem [14] these functions are steerable and need $M = 7$ basis functions. To measure the orientation along the contour, we use the phase independent squared sum of the output of the quadrature pair. This squared response as a function of the filter orientation ϑ at a point (x, y) represents an *oriented energy* $E^{(x,y)}(\vartheta)$. Because of the symmetry of the functions, the energy at every pixel is periodic with period π. To accurately estimate the *dominant* local orientation one could *pointwise* maximize the orientation energy by taking $\vartheta_{MAX}^{(x,y)} = \arg\max\{E^{(x,y)}(\vartheta) \mid \vartheta \in [0, \pi)\}$. However, to find this maximum value we do not search degree-wise for the maximum because there already exists an analytical solution for the maximization [14]. We further refer to the matrix of all these angular values $\vartheta_{MAX}^{(x,y)}$ corresponding to the image as $\mathbf{\Theta}$. Furthermore, there exists a separable basis set in Cartesian coordinates which considerably lowers the computational costs.

Computing the neural field input The previous section describes the theory and use of steerable filters. By means of those filters, we calculate both the matrix $\mathbf{\Lambda}$ describing a typical course of the head-shoulder-portrait and the matrix $\mathbf{\Theta}$ (computed from the image wherein a person is to be found) containing the dominant local orientation values.

Subsequently, we search for the presence of the *visual cue* head-shoulder-portrait, represented by the kernel $\mathbf{\Lambda}$, within the matrix $\mathbf{\Theta}$. To do this, we utilize a matching technique based on a *similarity measure* $m^{(x,y)}$. Due to the π-periodicity of the outcome of the steerable filters and in order to properly describe the likeness between two elements of $\mathbf{\Lambda}$ and $\mathbf{\Theta}$, the similarity function requires the same periodicity.

$$m^{(x,y)} = \frac{\displaystyle\sum_{i=0}^{I-1}\sum_{\substack{j=0\\ \lambda_{i,j}\neq 0}}^{J-1} \frac{1}{2}\left[\cos\left(2\left|\lambda_{i,j} - \vartheta_{MAX}^{(x+i-\frac{I}{2},y+j-\frac{J}{2})}\right|\right)+1\right]}{\mathrm{card}\left(\mathrm{supp}\left(\mathbf{\Lambda}\right)\right)} \qquad (2)$$

Herein, $\lambda_{i,j}$ refers to the element of $\mathbf{\Lambda}$ at position (i, j) and $\vartheta_{MAX}^{(x+i-\frac{I}{2},y+j-\frac{J}{2})}$ to the one of $\mathbf{\Theta}$ at $(x + i - \frac{I}{2}, y + j - \frac{J}{2})$. $I = J = 16$ represent the dimensions of the matrix $\mathbf{\Lambda}$. The normalization to the cardinality of the support of $\mathbf{\Lambda}$ (the support of a matrix considers only nonzero elements) ensures $m^{(x,y)} \in [0, 1]$ for the further processing. Figure 5 summarizes the processing steps.

2.2.2. *Skin color*

For the generation of a skin color training data set, portrait images of different persons of our laboratory were manually segmented. The images were acquired under appropriate lighting conditions typical for our laboratory environment. Since we want the color analysis to be unaffected by the presence of shadows or by changes

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

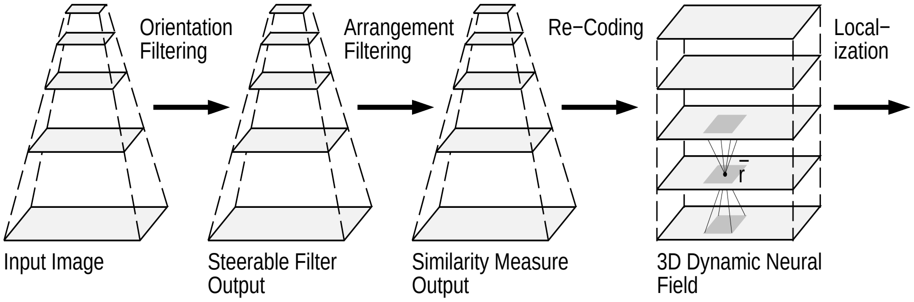

Fig. 5. Starting from a multi-resolution representation of the image, each level is treated by steerable filters. Applying the filter arrangement we determine a distance measure which is taken as input to a three-dimensional field of dynamic neurons. The resulting blob (locally delimited pattern of active neurons) is used to localize a person.

in illumination, this should allow to classify a point exclusively according to its hue and not to its intensity. In order to obtain almost constant color sensation, we first map the RGB color space into a fundamental color space and employ a color adaptation method (see [11]). Then we return in the RGB-image space and we consider the normalized (chromatic) color coordinates $r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$, and $b = \frac{B}{R+G+B}$. Since their sum is equal to 1, we can take any two of them to define the new color space (plane).

Representing our color data set in such a coordinate system, we obtain a color distribution as depicted in figure 6. Now we have to face the problem of modeling the color distribution by a probability density function. To density estimation given the finite number of data points, we consider a parametric method in which a specific form for the functional model is assumed. That model contains a number of adjustable parameters which have to be optimized to fit the model to the data set. Due to the form assumed by the real data distribution (see figure 6) and to its well-known properties, we decide to take as density function the d-dimensional multivariate normal probability function:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{3}$$

Here $\boldsymbol{\mu}$ is the d-dimensional mean vector, $\boldsymbol{\Sigma}$ the $d \times d$ covariance matrix, while the factor $(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}$ ensures that $\int_{-\infty}^{\infty} p(\boldsymbol{x})dx = 1$. Because $\boldsymbol{\Sigma}$ is symmetric it has $d(d+1)/2$ independent parameters. Considering also the additional d parameters of the mean vector $\boldsymbol{\mu}$, the density function is completely described by $d(d+3)/2$ parameters. Furthermore, after a person (face region) could be successfully localized, a new Gaussian model is created, more specific for the illumination and the skin type at hand. Via this model, the detection of skin colored regions, especially hands, can be improved. This is of special importance because the hand regions

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

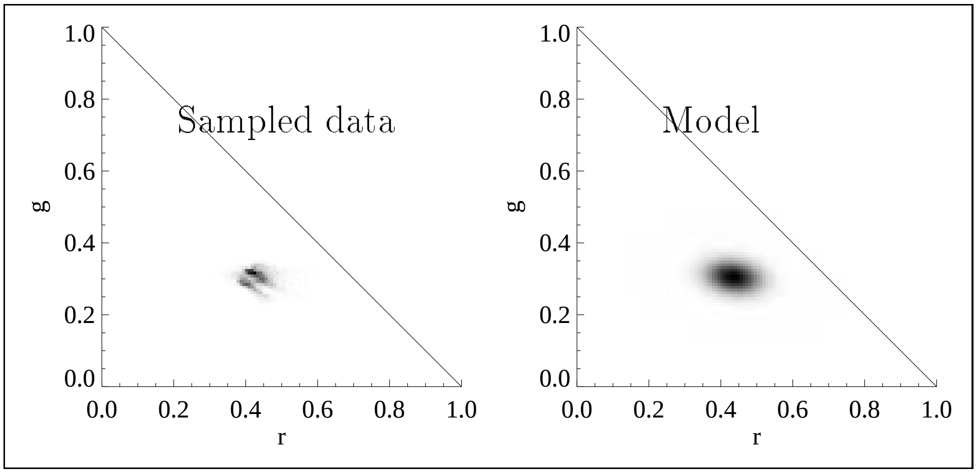

Fig. 6. Skin classification by means of a parametric model within the r-g-color space. On the left the histogram of the color sample data is depicted. For that data set the mean vector has the coefficients $\boldsymbol{\mu} = (0{,}43180; 0{,}30078)^{\mathsf{T}}$. The matrix of covariance for the data set has the following coefficients $\boldsymbol{\Sigma} = ((0{,}0016; -0{,}00013)^{\mathsf{T}}; (-0{,}00013; 0{,}00046)^{\mathsf{T}})$. On the right the statistical model with the above parameters is depicted.

cannot be segmented by structural information (see [15], and sec.).

A Mahalanobis-based distance measure is employed to compute the similarity between the color value of each pixel and the color model. To achieve an appropriate input for the 3D dynamic neural field, the resulting similarity map is recoded into an activity map, where the highest activity stands for the highest similarity.

2.2.3. *Facial structure*

We assume that a person is willing to interact with the system if her face is oriented towards the robot.
In our previous work, the detection of facial structure employed eigenfaces (see [10,6]). The disadvantage of that method is their computational complexity, resulting in time consuming calculations. Due to real-time constraints, a new, similar method was implemented. First, a prototype (mean) pattern of a frontally aligned face (15 x 15 pixels) was created by means of the images contained in the ORL data set (`http://www.cam-orl.co.uk/facedatabase.html`). Then we calculate the similarity between each image region and the prototype pattern via normalized convolution. The higher the convolution result, the higher the similarity, and the convolution result can be directly used as the input for the saliency pyramid. A related approach was also proposed by SIM ET.AL. [16], but differs in the used distance measure.

2.2.4. *The saliency pyramid as a 3D nonlinear dynamic field*

To achieve a good localization, a *selection mechanism* is needed to make a definite choice among those regions within the multi-scale pyramid where rather high similarity measures concerning the different cues are concentrated. Since dynamic neural fields are powerful for dynamic selection and pattern formation using simple homogeneous internal interaction rules, we adapted them to our purposes. In order to localize persons even at different distances, we use five fine-to-coarse resolutions in our scale space (see figure 3), as described above. Therefore, a neural field for selecting the most salient region should be three-dimensional, too. That field F can be described as a recurrent nonlinear dynamic system. Regarding the selection task, we need a dynamic behavior which leads to *one* local region of active neurons successfully competing against the others, i. e., the formation of one single blob of active neurons as an equilibrium state of the field. The following equations describe the system:

$$\tau\frac{d}{dt}z(\boldsymbol{r},t) = -z(\boldsymbol{r},t) - \mathrm{c}_h h(t) + \mathrm{c}_l\int_R w(\boldsymbol{r}-\boldsymbol{r}\,')y(\boldsymbol{r}\,',t)d^2\boldsymbol{r}\,' + \mathrm{c}_i x(\boldsymbol{r},t) \tag{4}$$

$$w(\boldsymbol{r}-\boldsymbol{r}\,') = 2\exp(\frac{-3|\boldsymbol{r}-\boldsymbol{r}\,'|^2}{2\sigma^2}) - \exp(\frac{-|\boldsymbol{r}-\boldsymbol{r}\,'|^2}{\sigma^2}) \quad , \tag{5}$$

$$y(\boldsymbol{r},t) = \frac{1}{1+exp(-z(\boldsymbol{r},t))} \quad \text{and} \tag{6}$$

$$h(t) = \int_R y(\boldsymbol{r}\,'',t)d\boldsymbol{r}\,'' \tag{7}$$

Herein $\boldsymbol{r} = (x,y,z)^T$ denotes the coordinate of a neuron, $z(\boldsymbol{r},t)$ is the activation of a neuron $\vec{r}$ at time t, $y(\boldsymbol{r},t)$ is the activity of this neuron, $x(\boldsymbol{r},t)$ denotes the external inputs (corresponding to the re-coded similarity measures for the different cues, combined by a Min-Max fuzzy operator), $h(t)$ is the activity of a global inhibitory interneuron, $w(\boldsymbol{r}-\boldsymbol{r}')$ denotes the Mexican-hat-like function of lateral activation of neuron $\boldsymbol{r}$ from the surrounding neighbourhood $N \subseteq \mathbb{R}^3$. For one $\boldsymbol{r}$, N is symbolically marked as dark regions in figure 5 (right). Further, τ is the time constant of the dynamical system and σ is the deviance of the gaussians determining the function of lateral activation. For the computation we used the following values for the constants: $c_h = 0.025$, $c_l = 0.1$, $c_i = 0.1$, $\sigma = 2$ (halved z-direction), $\tau = 10$ with $\Delta T = 1$ (ΔT: sampling rate). The range R of the function of lateral activation reachs over 5 pixels and 3 pixels in z-direction, respectively (anisotropic neighbourhood).

As also illustrated in figure 5, to use a three-dimensional neural field, we have to consider the local correspondences between the resolution levels. Therefore, we apply a re-coding into a cuboid structure. One side effect is that the coarser a pyramid level is the less we can locate something by means of the similarity measure. However, without particularly treating this effect we just noticed that those levels z of the neural field activated from the rather coarse pyramid levels take a little few more steps to develop a blob (or a part of a blob, respectively).

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

2.3. *Results of the multiple-cue approach*

The results of the multiple-cue approach are qualitatively illustrated in figure 7. The images of the rightmost column show the state of three layers of the dynamic neural field in a snapshot at that moment when the activity change of the most active neuron became less than 1%. On average, the system takes 11 iteration steps using a time-discrete Euler method. The range of the blob is not restricted to one plane. To get a more precise specification of the distance of a person one could interpolate the z-coordinate of the blob center within the field.

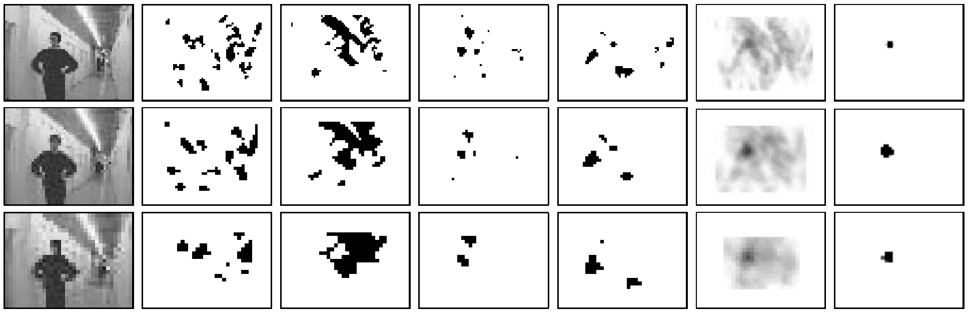

Fig. 7. Localization results in an indoor environment (middle three layers of the multiscale representation): The localization of a person occurs not sharply at one of the pyramidal planes, the originating spatial blob (rightmost column) is most strongly developed on the central of the five planes. Each row contains the results of one of the five (distance $1/\sqrt{2}$) computed resolution steps. The seven columns depict the following: input, results of the orientation filtering for selected angles 0°, 45°, 90° and 135°, the result of the filtering with the filter arrangement and finally the result of the selection within a three-dimensional field of dynamic neurons.

The novel approach with a three-dimensional dynamic neural field can be assessed as a robust method for the selection process.

To emphasize finally the performance of the multiple-cue approach, figure 8 shows typical results in a highly structured indoor environment.

Furthermore, what we really need is a most possible non-ambiguous person localization which can only be ensured if only one correct localization is achieved, but, unfortunately, this is not often the case. In general we have a (very limited) number (2-5) of candidate regions (see figure 8), and one of these candidate regions typically covers the person we look for. To make the final selection we use both, the motion-based segmented image provided by the omnidirectional camera which covers the whole surroundings of the robot, and the result of the multiple-cue approach.

After a person could be successfully localized a gesture recognition process is started. The remainder of the paper focuses on that gesture recognition process.

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

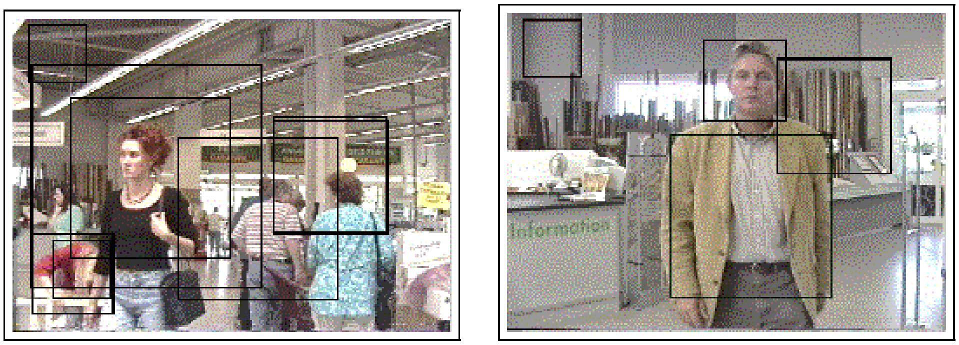

Fig. 8. Person localization with the previously described multiple-cue approach in a highly structured indoor environment.

3. View-based Posture Description

Throughout this paper the following definitions are considered.

Definition 1 [Posture/Pose] *A posture or pose is a couple determined by the only static hand locations with respect to the head position. The spatial relation of face and hands determines the behavioral meaning of each posture.*

Definition 2 [Gesture] *A gesture is a series of postures over a time span connected by motions.*

3.1. *Posture segmentation*

The segmentation of face and hands as the gesture relevant parts is exclusively based on skin color processing. Therefore, we presuppose that skin color is always present within an image. Segmentation is a decision process where we have to decide whether a pixel belongs or not to the hands or the head of the user. Obviously after this decision a significant amount of information is lost because there is no way to infer the original image content from the segmented image.

After detecting the location of the head as described in the previous chapter, we consider a region of interest around it which we call *head box* (figure 9, left). Then we characterize the distribution of the pixel values inside that region of interest by a multidimensional Gaussian with centroid location and covariance matrix describing the local distribution around the centroid. By doing that, we adapt the skin color model to fit more specific for the illumination and the skin type at hand. Therefore the detection of skin colored regions can be improved. We handle multiple scales by choosing head boxes of different sizes according to the level in the pyramide.

By using the chromatic projections r and g (see section 2.2.2) of each pixel inside the head box, the actual color model is uniquely determined by the multivariate normal density of equation 3 where the mean $\boldsymbol{\mu}$ is a two-dimensional vector and $\boldsymbol{\Sigma}$ is a 2×2 covariance matrix. Using the Mahalanobis distance from $\boldsymbol{x}$ to $\boldsymbol{\mu}$, any pixel $\boldsymbol{x}$ of the image is then classified to be or not a member of the skin class according to an empirically determined threshold value (figure 9, middle).

In order to reshape the segmented regions we filter the binary image by using the *median filter*. In case of binary image this correspond to just set the central mask point with the value which appears more times in the neighborhood. On binary images, median filters act as *dilation operator* as well as *erosion operator* according to the number of on-values around the actual pixel and the mask size. The filter fills small holes or cracks and smoothes the contour line of the regions while removes small regions originated by outliers.

Assuming the hands and head regions to correspond to the three greatest ones we need a process for extracting them. The selection process is executed by a winner-take-all (WTA) neural network. Beginning from the winning neuron the underlying region is segmented and its area is computed. When the measure of this area is less than a fixed percentage, called *percentage threshold* of the largest hitherto segmented region, it is considered as a noisy region and the selection task is stopped. Otherwise, the region is inhibited and a new competitive process begins. In any case, the selected region has to be permanently inhibited in order to avoid that it is selected again in the next competition task. Without inhibition the currently selected region would be always selected again (it is the largest!) and would not permit smaller regions to win the competition. We can see that according to the choice of the percentage threshold may happen that the WTA selects less than three regions. Anyway, at least one region is always selected.

Further we determine the centers of gravity of the selected regions and we model each of them separately as a circle around its centroid. Now one problem arises. How should we choose the radius for the circles? The first idea is to calculate the multivariate normal density for every region and to use the lengths of the principal axes of the associated ellipsoid as radius. Although this seems to be well-founded, it turns out not to be.

Let us make a step back to the color segmentation task. If we had chosen a higher (lower) threshold value during the segmentation, the shape of the remaining regions would be now bigger (smaller). Moreover even totally different regions could have been selected. A different threshold value means different multivariate normal densities, and, consequently, different principal axes values. To avoid ambiguous posture modeling, the solution is simply to use an arbitrarily constant value for the radius (see figure 9, right).

3.2. *Feature extraction*

Frequently, the features extracted from the input data which are invariant under the requested transformations base on *moments* of the original data.

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

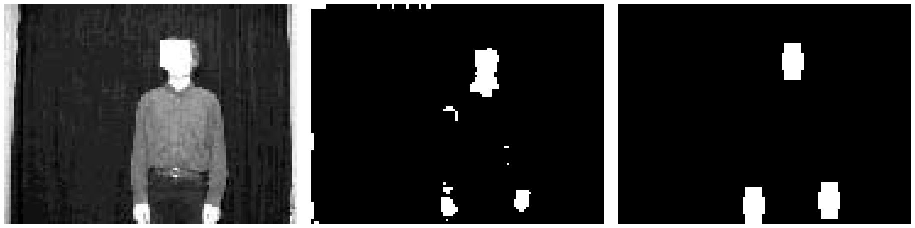

Fig. 9. From left to right: head localization result; thresholded skin classification by means of an adapted color model derived from the pixel distribution around the head location, and finally, modeling of the hand and head regions as circle around their centers of mass.

Given an arbitrary two-dimensional function $f(x,y)$ and two integer numbers p and q, the $(p+q)$th order *regular moment* m_{pq} is defined by

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \tag{8}$$

For our purpose, we consider a finite image plane and therefore the integrals are over that finite surface. In that case, they have to be replaced by discrete sums, so that equation 8 becomes

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{9}$$

It has been proved [17] that, when the function is sufficiently well behaved mathematically, by using a sufficiently large number of moments we merely obtain a different but complete description of it. The moments constitute the coefficients in a series expansion of some complete figure description. Thus, every function f is uniquely determined by, and uniquely determines the set of the infinite moments m_{pq}. Considering only a subset of the moment set, only a partial function reconstruction can be obtained.

The calculation of the moments furnishes a systematic procedure for extracting a set of features from an image, i. e., the extraction of features from the original input data which are invariant under some given transformations as for example translation, scaling, rotation, reflection, and so on.

By centering the x and y axes at the centroid of the function f, we can define its $(p+q)$th order *central moment* $\boldsymbol{\mu}_{pq}$ as

$$\boldsymbol{\mu}_{pq} = \sum_x \sum_y x^p y^q f(x-\bar{x}, y-\bar{y}) \tag{10}$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$ represent the x and y coordinates of the image centroid, respectively. Although the central moments are invariant under translation in their continuous form, the use of that moments in discrete form gives only an approximate translation invariance due to the edge effects and the finite sums.

Applying the theory about algebraic invariants [18] we can extend the invariance properties of the central moments by making them invariant also to scaling. Thus we obtain the $(p+q)$th order *normalized central moment* ν_{pq}

$$\nu_{pq} = \frac{\boldsymbol{\mu}_{pq}}{\boldsymbol{\mu}_{00}^{(1+\frac{p+q}{2})}} \tag{11}$$

It is easy to verify that they remain simultaneously unchanged under image translation and size changes.

The computation of these moments for binary image yields theoretically an error-free estimate of the continuous moments which is also independent of illumination as opposed to the value deriving from greyvalue images.

3.2.1. *Choice of the invariants*

From the binary posture model (figure 9,c) we compute one feature vector $\boldsymbol{v}$ containing 15 translation and scale invariant elements. The aim is to characterize as well the shape of the segmented scene as the spatial relations among the regions within it.

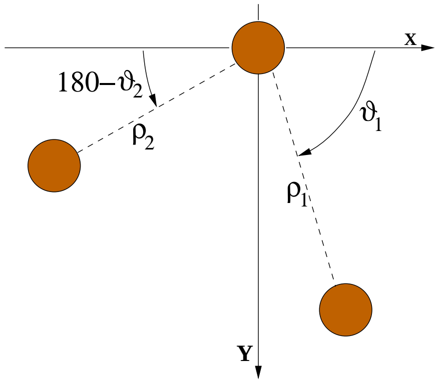

Fig. 10. The new defined coordinate system with origin centered in the centroid describing the head region. The two points (ρ_1, ϑ_1) and (ρ_2, ϑ_2) represent the polar coordinates of the left and right hand centroids.

As first invariant values we take seven of the ten normalized central moments up to the third order. We drop out as well the zeroth order moment ν_{00} as those of the first order because they always assume value one and zero, respectively.

The computation of the remaining feature vector elements is carried out with the goal to compensate the shift variation of the person gesticulating in front of the camera. Thus, we choose for each image a suitable coordinate system by fixing its origin point at the current determined head's center of mass. That allows to calculate the other feature components relating to the head position and regardless to the position of the user within the image. In this new coordinate system, we use the polar coordinates of both hands's centers of gravity (figure 10) and the normalized Euclidean velocities of the hand centroids along both the x and y axes, in order to ensure invariance also with respect to image size changes. The chosen features are listed and described in table 11.

Feature Nr.	Symbol	Description
$1 \ldots 7$	$\nu_{pq} \mid p, q = 1 \ldots 3$	Normalized central moments of second and third order.
8,9	ϑ_1, ϑ_2	Values in the range $[-180, 180]$ indicating the angle expressed in degree between the x axes and the segment connecting the origin with the centroid of the right and the left hand, respectively (see figure 10).
10,11	ς_1, ς_2	Let ρ_1 and ρ_2 indicate the length of the segments connecting the origin with the centroid of the right and the left hand, respectively (see figure 10); these value are defined as $\frac{\rho_1}{\max\{\rho_1,\rho_2\}}$ and $\frac{\rho_2}{\max\{\rho_1,\rho_2\}}$, respectively.
12,13	v_{Rx}, v_{Ry}	With the above definition these values represent the Euclidean velocity of the right hand centroid normalized by $\max\{\rho_1, \rho_2\}$.
14,15	v_{Lx}, v_{Ly}	As the previous two features but regarding centroid of the left hand.

Fig. 11. Description of the components of the feature vector.

It is worth to notice that the normalizing factor $\max\{\rho_1, \rho_2\}$ used in the definition of the last six features (from number 10 to number 15) of table 11 is required to ensure size invariance. Moreover, we can see that the two invariants ς_1 and ς_2 always assume values within the range $[0, 1]$ and at least one of them is exactly 1.

3.2.2. *Whitening rescaling*

Figure 12 shows the values of the feature components from a segmented binary image. As we can see the single components have values which differ in a significant manner also by some orders of magnitude. Because that size dissimilarity does not reflect the relative importance of the individual components, it is useful to rescale them.

There are a lot of techniques for data normalization or rescaling. In our work, we perform a linear rescaling known as *whitening* which allows for correlation among

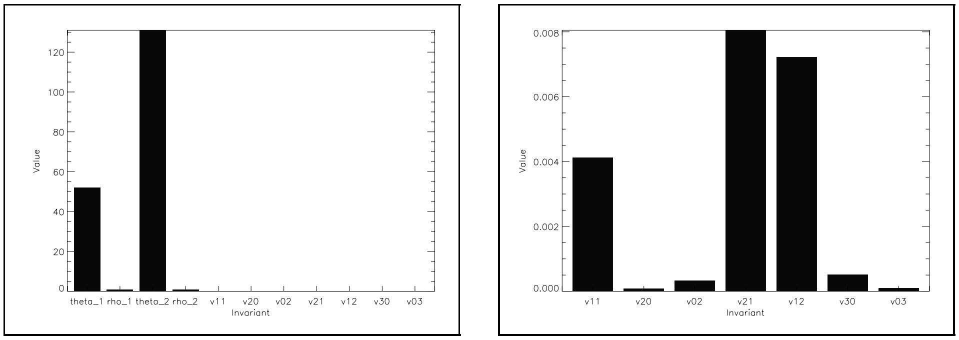

Fig. 12. Left: First 11 out of the 15 elements of one feature vector are depicted but only the first four are visible due to their different order of magnitude. The depicted vector assumes the component values: (52, 1, 127, 0.98,0.00413,0.0000829,0.000329,0.0000805, 0.00721,0.000516,0.0000977). Right: the feature values which could not have been represented in the left table are now visible by using a different scale.

the variables considered. From the training set made up of N feature vectors $\{\boldsymbol{f}_1 \ldots \boldsymbol{f}_N\}$ we calculate component-wise the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ with respect to the data point of the training set. Let us consider the eigenvector equation for the symmetric matrix $\boldsymbol{\Sigma}$ in matrix notation

$$\boldsymbol{\Lambda} = \boldsymbol{U}^\top \boldsymbol{\Sigma} \boldsymbol{U} \tag{12}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix whose elements consist of the eigenvalues of $\boldsymbol{\Sigma}$, while $\boldsymbol{U}$ is a matrix whose columns consist of its eigenvectors and satisfies the condition $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}$. The proper whitening transformation acts on any input variable $\boldsymbol{x}$ as follow

$$\tilde{\boldsymbol{x}} = \boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{U}^\top (\boldsymbol{x} - \boldsymbol{\mu}) \tag{13}$$

The matrix $\boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{U}^\top$ is called *whitening matrix*. In that transformation the data distribution is whitened by means of its eigenvectors together with the corresponding eigenvalues. In the new coordinate system the data set has zero mean and unitary covariance matrix.

4. Stochastic Recognition with HMM's

Hidden Markov Models [19,20] are probabilistic finite state machines well-suited in dealing with the statistical and sequential nature of time-varying input patterns. They are the basis for a lot of applications especially in the field of speech recognition [21,22,23], hand-writing recognition [24,25], and recognition of gestures [26,27].

A HMM consists of a finite number of states connected one another by directed arcs according to a predefined topology. Each arc is associated with one probability value that is called state transition probability. At regular time intervals the model

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

undergoes a change of state according to the set of transition probabilities. Also a change back to the same state is possible. Each state computes the estimation of the likelihood for a certain input observation vector by means of a probability density distribution function which can be discrete or continuous. After defining also an initial state distribution, the HMM can be used as a generator of sequences of observations or as a model for an observation the sequence is generated by.

There are two concurrent stochastic processes associated with each HMM: a set of state output processes that models the local stationary character of the observation at each time step, and the state sequence that models the temporal structure of the signal being modeled. Because this latter state sequence is not directly observable the Markov model is called *hidden*.

In our work, we used as many HMM's as the number of gestures to be detected. The training and decoding of the models are based on the posterior probability $P(M|\boldsymbol{X}_0^t)$ that the feature vector sequence $\boldsymbol{X}_0^t = \boldsymbol{X}_0\boldsymbol{X}_1 \ldots \boldsymbol{X}_t$ has been produced by the model M. In the learning phase, the set of parameters maximizing that probability are sought for every sequence $\boldsymbol{X}_0^t$ associated with the model M. This strategy is referred to as the *maximum a posteriori* criterion. During the recognition stage, given an observation sequence $\boldsymbol{X}_0^t$ and a fixed set of parameters, the goal is to find out that model M among many models that maximizes $P(M|\boldsymbol{X}_0^t)$.

Unfortunately, the learning process generally does not consent (Andrea?) to expressly characterize $P(M|\boldsymbol{X}_0^t)$ but permits the characterization of the probability $P(\boldsymbol{X}_0^t|M)$ that a given model generates certain feature sequences. Using the Bayes' rule, one can express $P(M|\boldsymbol{X}_0^t)$ in terms of $P(\boldsymbol{X}_0^t|M)$ as

$$P(M|\boldsymbol{X}_0^t) = \frac{P(\boldsymbol{X}_0^t|M)P(M)}{P(\boldsymbol{X}_0^t)} \tag{14}$$

where $P(M)$ is the prior probability of the model, $P(\boldsymbol{X}_0^t)$ is the prior probability of the vector sequence, and $P(\boldsymbol{X}_0^t|M)$ is referred to as the *likelihood* of the data given the model. Because we suppose each gesture to be the same prior probability, $P(M)$ is a constant term. In addition since $P(\boldsymbol{X}_0^t)$ can be assumed constant because it does not depend on the models, the estimation of equation (14) is equivalent to calculating only the likelihood $P(\boldsymbol{X}_0^t|M)$. In that case, when the training criterion aims at the maximization of the quantity $P(\boldsymbol{X}_0^t|M)$, it is referred to as *maximum likelihood* criterion. This is exactly the learning strategy we adopt.

5. Hybrid SOM/DHMMs for Gesture Recognition

5.1. *Self-organizing maps for symbol production*

The goal of the posture analysis is the extraction of local features along the hand trajectory, yielding a sequence of time ordered multi-dimensional feature vectors. The further step is concerned with the quantization of these feature vectors into a sequence of symbols.

A Self-Organizing Map (SOM) [28] is used to preserve the topology of the high-dimensional feature space by mapping the feature vectors onto a two-dimensional space. Due to the sequential nature underlying each gesture such a topology-preserving map can be exploited to constitute trajectories where the SOM best-matching neurons are recorded during the process. A similar approach has been used by Waldherr [27].

The SOM clusters the unlabeled training feature vectors which lie near one another in the feature space. During the training phase as well the codebook vector most sensitive to the actual training vector as those in its (variable) neighborhood are tuned maintaining a well-balanced set of weight values with respect to the input density function.

The weight adjustment is carried out using the Euclidean distance between the actual multi dimensional input vector and the connecting weight vectors, a time-dependent learning rate, and a neighborhood function that decays like the Gaussian probability density function when the topological distance between the best-matching unit and the actual vector increases.

We start the learning process with a large radius covering all the units in order to prevent the formation of undesired outliers in the clustering due to the limited training data set. Our SOM has 800 nodes organized in a 40×20 grid. The feature vectors are 15-dimensional and the SOM is trained by decreasing the neighborhood radius from 6 to 1 and the learning rate from the value 0.9 to 0.

After the clustering process, each neuron of the network corresponds to a cluster in the input feature space. Proceeding from the self-organizing process we tune the weight vectors using the unsupervised Learning Vector Quantization (LVQ) method causing the weights to approach the decision boundaries [28].

In order to utilize the SOM for classification, we divide each gesture of our vocabulary in *subgestures* or *posture classes* and we label each of them with a different symbol (see figure 13 for the hand-waving-right movement). We divide the gestures of our vocabulary into altogether 32 subgestures (9 for each left-,right-waving; 5 for each go left/right; 4 for stop). For class discrimination purposes we hand-label each SOM cluster. These labels were assigned to the units according to the subgesture subdivision as depicted in figure 13 by using the recorded training samples as input.

5.2. *Using DHMMs for classification*

In subsection 5.1 we assigned each feature vector to a symbol which corresponds to a codeword in the codebook created by LVQ. The feature vectors of the data set for training were vector quantized. The need of a vector quantizer to map the continuous observation vectors into discrete symbols arises from the choice to use DHMM's as recognizer.

For the choice of the model topology, there is no theoretically way to rely on. The choices we made depend on the gesture being modeled. For each movement to be detected, we create one left-to-right DHMM (figure 14) with as many states as

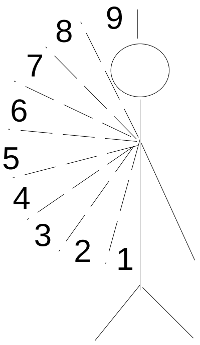

Fig. 13. Waving-right gesture hand-labeling. That movement is divided in 9 subregions each covering exactly 20 grad of the two-dimensional plane surface the gesture is projected on. Each subregion is labeled by one symbol.

the subregions which this gesture is divided in. In such a model, each DHMM state is associated with a single movement's subgesture (figure 13).

In the learning phase, the parameters of each DHMM are optimized so as to model the training symbol sequences from the corresponding gesture. More precisely, the parameter of each model are estimated with symbol sequences of the according gesture samples applying the Baum-Welch training algorithm [19]. The latter is an iterative procedure based on the Maximum Likelihood criterion aiming at maximizing the probability of the samples given the model at hand and can be considered as a form of the *expectation-maximization* algorithm [29].

Because we consider a gesture as a sequence of subgestures the recognition process consists in comparing a given sequence of symbols with each DHMM. That gesture associated with the model which best matches the observed symbol sequence is chosen as the recognized movement.

6. Continuous HMMs for Automatic Gesture Recognition

Up to this point, we have considered the case when the observations were characterized as discrete symbols from a finite alphabet. In this situation, we could use only discrete probability density functions within each model state. The main problem with this approach is the need to quantize the continuous feature vectors via codebooks. Because that quantization process might be accompanied by distortion or loss of information, it could be advantageous to utilize the HMMs with continuous observation density functions. In this case, these density functions are some parametric probability distributions or mixtures of them.

The most common parametric distribution used is the mixture of Gaussian density which can be expressed for a generic state i as

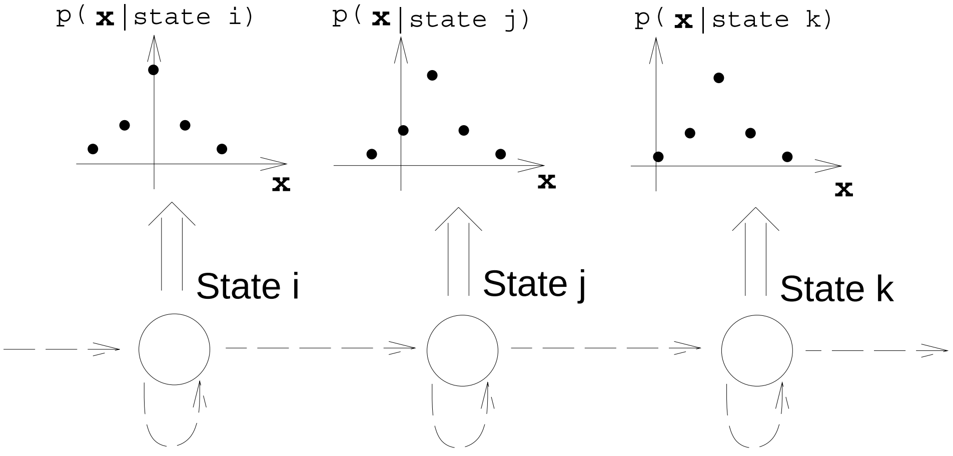

Fig. 14. Left-to-right DHMM. This model is called left-to-right or Baskis model because it has the property that as time increases the state changes proceed from left to right. The dashed arrows depict the transition probabilities among the states. Here only transitions from a state to the next one or to itself are allowed. The probability distribution functions assume discrete values.

$$p_i(\boldsymbol{X}) = \sum_{m=1}^{M} c_{im}\mathcal{N}(\boldsymbol{X}, \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) \tag{15}$$

where M is the number of mixtures ($M = 3$ in our experiments), $\boldsymbol{X}$ is the vector being modeled, c_{im} is the mixture coefficient for the m-th mixture in state i and $\mathcal{N}$ is any strictly log-concave or elliptically symmetric density function with covariance matrix $\boldsymbol{\Sigma}_{im}$ and mean vector $\boldsymbol{\mu}_{im}$ in state i for the m-th mixture.

With D-dimensional data (here $D = 15$ is the dimension of the feature vectors) and using the Gaussian function as parametric probability distribution, the function $\mathcal{N}(\boldsymbol{X}, \boldsymbol{\mu}_{\rangle\Updownarrow}, \boldsymbol{\Sigma}_{\rangle\Updownarrow})$ in equation (15) can be expressed as

$$\mathcal{N}(\boldsymbol{X}, \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) = \frac{e^{(-1/2(\boldsymbol{X}-\boldsymbol{\mu}_{im})^{\top}\boldsymbol{\Sigma}_{im}^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_{im}))}}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_{im}|^{1/2}} \tag{16}$$

As the dimension of the feature vectors increases, as well the length of the mean vectors as the size of the covariance matrices becomes greater. But while the increase in size of the mean vectors is proportional to the one of the observation vector, the enlargement in size of the covariance matrices is even square proportional to the vector dimension. Hence, with multi-dimensional observation vectors, the number of parameters of the mixture of Gaussian is very large and its estimation becomes computationally expensive. Additionally, with insufficient training data to estimate some of these parameters will assume more or less arbitrary values.

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

To avoid a huge number of parameters and, at the same time, to have representative models, we approximate the covariance matrices by diagonal matrices, and we tie it over the whole model. Under that simplification the model parameters can be estimated faster by using the Baum-Welch learning algorithm [19] again.

7. Preliminary Results

To train and test each HMM in both discrete and continuous case, we gathered the data from four people performing five repetitions of the gesture to be described. The categories to be recognized are five. Therefore, we take the same number of left-to-right HMM's each corresponding to one class.

Table 1. Recognition results using DHMM's.

Gesture	% of not classified patterns	% of false classified patterns	Recognition rate in %
stop	9.2	13.2	77.6
waving right	8.4	11.1	80.5
waving left	8.7	10.0	81.3
go right	9.6	8.6	81.8
go left	10.2	9.6	80.2

The sequences were captured by a color camera at a frequency of 25 frames per second and digitized into 120×90 pixel RGB images. Table 1 summarizes the achieved performance concerning the recognition task by utilizing a recognizer based on the SOM/DHMM hybrid architecture; Table 2 shows the recognition performance achieved by using only CHMMs.

Table 2. Recognition results using CHMM's.

Gesture	% of not classified patterns	% of false classified patterns	Recognition rate in %
stop	10.4	10.0	79.6
waving right	7.3	10.3	82.4
waving left	8.8	8.5	82.7
go right	7.4	7.8	84.8
go left	8.1	8.0	83.9

We consider an input as not classified if after feeding it into each HMM either the difference between the highest and the second highest output is not over an heuristically determined threshold or if all the outputs are under a given threshold.

Compared with continuous models, discrete distributions normally require less parameters. This means that DHMM's have less memory requirements and need less training data to achieve good generalization performance. Moreover, discrete models require shorter recognition and training time since they do not have to calculate any mixture of Gaussian distribution. For discrete models only quantization of the observation vectors has to be performed while the state probability estimation is replaced with a look-up table.

From a direct comparison of the recognition rates regarding our problem, we can see how the CHMM-based system leads to slightly better results than the hybrid SOM/DHMM-based one. We think that this is mainly due to the continuous intrinsic character of the feature vectors. The conversion of them into discrete symbols via vector quantization can worsen the recognition task. In spite of our experimental results, we do not state that CHMM's outperform SOM/DHMM-based recognizers in general. Due to the limited training data it would be a shaky conclusion, strongly dependent on the implementation and the few data at hand.

Anyway, the recognition rate of both systems can be improved by using a discriminative training algorithm instead of the Baum-Welch algorithm giving arise to a poor discriminative power among different models.

8. Conclusions and Outlook to Future Work

Besides the performance concerning posture recognition, the person localization is the most crucial but absolutely necessary prerequistite for the function of the whole system. The use of multiple cues and their integration into a selection process via 3D dynamic neural fields led to a satisfying person specific saliency system. Using a Chugai Boyeki CD 08 video camera with maximum wide angle mode, the multiscale representation covers a distance from 0.5 to about 2.5 meters. Within this interval, the localization is very robust against slight rotations (up to 15°), scene content, and illumination. Furthermore, the integration of the omnidirectional camera makes it easier for the system to detect a person it its surroundings, which significantly speeds up the localization process.

So far, both methods proposed for gesture recognition were tested on a small set of simple gestures and thus have very limited scope. We are currently extending both systems in order to overcome this limitation. The aim is to design a system that can work with a larger vocabulary of gestures, and remain user independent. The performances of the two architectures depend strongly on the number of training pattern and also how well that patterns are representative for each class. It means that the training patterns have to cover the maximum test pattern range as possible.

On the one hand HMM's provide a good representation of the sequential nature of the human movements, on the other they suffer from several limitations and drawbacks because of the assumptions exploited for the implementation of their learning and decoding algorithms [23]. We refer, for example, to the strong statistical assumption that the probability density functions associated with the states can be described by a fixed parametric function. Again, it is supposed every state change to depend only on the current and previous state and not on all the predecessor

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

ones (*first-order HMM*). Also the likelihood of an observation vector is assumed not to depend on the previous observations but only on the current state (*context-independent* assumption).

In addition, HMMs consider the sequence of feature vectors as a piecewise stationary process. Hence, even though gesticulating is a non-stationary process, we have to assume that over a short period of time the statistics of the movement underlying the gesture do not differ from sample to sample neglecting the correlations between successive feature vectors (*statistical time-independence* of the observation vectors).

HMM's trained with the non-discriminative Baum-Welch algorithm show also poor discriminative capability among different models. Namely, by maximizing the maximum likelihood instead of the maximum a posteriori, the HMMs are trained only to generate high probabilities for its own class and not to discriminate against models.

Due to their inherently discriminant nature and lack of distributional assumptions we are currently using and testing a system with neural networks to estimate the probability for HMM states.

The overall system has to be understood as work in progress, undergoing continuous changes. Currently, the major constraints are that the gesture recognition process does not work in real-time, whereas the localization process does, and that the posture segmentation uses only skin color which causes problems when other skin-colored objects are in the scene. The latter problem is currently to be eliminated by additionally using motion information, resulting in a search for moving skin color.

In the long run, we want to develop a continuous action-perception cycle between the robot and its human user in service system domains, where the architecture described here could be one building block.

Acknowledgments

This work was supported by the TMR Marie Curie Research Training Grant # ERB FMBI CT 97 2613 to A. Corradini.

References

[1] Darrell, T., Basu, S., Wren, C., and Pentland, A. Perceptually-driven Avatars and Interfaces: active methods for direct control. In *SIGGRAPH'97*, 1997. M.I.T. Media Lab Perceptual Computation Section, TR 416.

[2] Kahn, R. *Perseus: An Extensible Vision System for Human-Machine Interaction.* PhD thesis, University of Chicago, 1996.

[3] Kortenkamp, D., Huber, E., and Bonasso, P.R. Recognizing and interpreting gestures on a mobile robot. In *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996.

[4] Corradini, A., Braumann, U.-D., Boehme, H.-J., and Gross, H.-M. Contour-based person localization by 3dneural fields and steerable filters. *Proceedings of the IAPR*

Workshop on Machine Vision Applications (MVA '98), pages 93–96, 1998.

[5] Wren, C., Azarbayejani, A.and Darrell, T., and Pentland, A. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997. M.I.T. Media Lab Techreport TR 353.

[6] Boehme, H.-J., Braumann, U.-D., Brakensiek, A., Krabbes, M., Corradini, A., and Gross, H.-M. User Localisation for Visually-based Human-Machine-Interaction. In *International Conference on Automatic Face- and Gesture Recognition*, pages 486–491. IEEE Computer Society Press, 1998.

[7] Boehme, H.-J., Braumann, U.-D., Brakensiek, A., Krabbes, M., Corradini, A., and Gross, H.-M. Neural Networks for Gesture-based Remote Control of a Mobile Robot. In *International Joint Conference on Neural Networks*, volume 1, pages 372–377. IEEE Computer Society Press, 1998.

[8] Amari, S. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[9] K. Kopecz. Neural field dynamics provide robust control for attentional resources. In *Aktives Sehen in technischen und biologischen Systemen*, pages 137–144. Infix-Verlag, 1996.

[10] Boehme, H.-J., Brakensiek, A., Braumann, U.-D., Krabbes, M., and Gross, H.-M. Neural Architecture for Gesture-Based Human-Machine-Interaction. In *Gesture and Sign-Language in Human-Computer Interaction*, Lecture Notes in Artificial Intelligence, pages 219–232. Springer, 1998.

[11] Pomierski, T. and Gross, H.-M. Biological Neural Architectures for Chromatic Adaptation resulting in Constant Color Sensations. In *ICNN'96, IEEE International Conference on Neural Networks*, pages 734–739. IEEE Press, 1996.

[12] Jones, J.P. and Palmer, L.A. An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex. *Journal of Neurophysiology*, 56(8):1233–1258, 1987.

[13] Koenderink, J.J. and van Doorn, A.J. Receptive Field Families. *Biological Cybernetics*, 63:291–297, 1990.

[14] Freeman, W.T. and Adelson, E.H. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9):891–906, 1991.

[15] Hunke, M.H. Locating and Tracking of Human Faces with Neural Networks. Technical report, Carnegie Mellon University Pittsburgh, 1994. CMU-CS-94-155.

[16] Sim, T., Sukthankar, R., Mullin, M., and Baluja, S. High"=Performance Memory"=based Face Recognition for Visitor Identification. Technical report, Carneghie Mellon University, Institute of Computer Science, 1999.

[17] Duda, R.O. and Hart P.E. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[18] K. Hu M.˙ Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, pages 179–187, February 1962.

[19] Baum, L. and Petrie, T. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, January 1966.

[20] R. Rabiner L.˙A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

[21] Bahl L.R., Brown P.F., de Souza P.V., and Mercer R.L. Maximum mutual information estimation of hidden markov model parameters for speech recognition. *Proceedings of the IEEE Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP), Tokyo*, pages 49–52, 1986.

[22] J.K. Baker. The dragon system - an overview. *IEEE Transaction on Acoustics, Speech,*

Int. Journal on Artificial Intelligence Tools (IJAIT), vol. 9 (2000), no. 2, pp. 177-204

and Signal Processing, 23(1):24–29, 1975.

[23] H. Bourlard and N. Morgan. Connectionist speech recognition. *Kluwert Academic Publishers, Dordrecht, The Nederlands*, 1994.

[24] Cho S.-B. A hybrid method of hidden markov model and neural network classifier for on-line handwritten character recognition. *Kohonen T., M"akisara K., Simula O. and Kangas J. ed., Proceedings of the 1991 International Conference on Artificial Neural Networks*, pages 741–744, 1991.

[25] Amlan Kundu. *Handbook of Character recognition and Document Image Analysis*, chapter Handwritten word recognition using Hidden Markov Model, pages 157–182. World Scientific Publishing Company, 1997.

[26] Hienz, H., Bauer, B., and Kraiss, K.-F. HMM-Based Continuous Sign Language Recognition Using Stochastic Grammars. In *GW'99 - The 3rd Gesture Workshop, Gif"=sur"=Yvette, France*, pages 185–196. Springer, 1999.

[27] Waldherr, S., Thrun, S., and Romero, R. A gesture-based interface for human-robot interaction. *Autonomous Robots*, page to appear, 2000.

[28] T. Kohonen. *Self-Organizing Maps.* Springer Verlag, 2nd Edition, 1997.

[29] Dempster,A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1(B 39):1–38, 1977.