Improving Adaptive Skin Color Segmentation by Incorporating Results from Face Detection

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

Abstract

The visual tracking of human faces is a basic functionality needed for human-machine interfaces. This paper describes an approach that explores the combined use of adaptive skin color segmentation and face detection for improved face tracking on a mobile robot. To cope with inhomogeneous lighting within a single image the color of each tracked image region is modeled with an individual, unimodal Gaussian. Face detection is performed locally on all segmented skin-colored regions. If a face is detected, the appropriate color model is updated with the image pixels in an elliptical area around the face position. Updating is restricted to pixels that are contained in a global skin color distribution obtained off-line. The presented method allows to track faces that undergo changes in lighting conditions while at the same time it provides information about the attention of the user, i.e. whether the user looks at the robot. This forms the basis for developing more sophisticated human-machine interfaces capable of dealing with unrestricted environments.

I. Introduction

In recent years many researchers have started to develop advanced human-machine interfaces for various applications [1]. For mobile robots two cues are frequently used: speech and gesture. While gesture commands are often limited to giving simple motion commands or pointing at certain objects (see e.g. [2], [3]) the use of natural language allows a richer interaction with a robot (e.g. [4]). However, if the microphone is mounted on the robot instead of being positioned close to the humans mouth (e.g. by using a wireless headset), there may be speech from several people present in the recorded signal.

In human-human communication the face of the communication partner is an important source of information: We can recognize *who* is talking to us and *how* he is talking to us, i.e. the emotional state of the dialog partner. We can even deduce *when* somebody is talking to us, as this usually is coupled with the person looking at us. Consequently, in a multi-modal human-robot interface the speech recognition algorithm can be triggered by a face detection algorithm to process speech input only if a person is looking at the robot while speaking. Given the recognized face position of the person a beamforming microphone array can be used to filter out other signal sources not originating from the speaking person [5]. Vice versa, microphone arrays can be used to localize a speaker (cf. e.g. [6]).

While face detection can be used for controlling a speech recognizer, it is in general computationally expensive to perform. To limit the search area for face detection, a preprocessing step based on color can be used (see e.g. [7]), as a face is always skin-colored. However, color is not a stable feature as a mobile robot encounters varying lighting conditions while moving around. This applies not only to the color of a face, but similarly to the color of hands, which are usually the basis of approaches demonstrating the instruction of robots by human gestures [2], [3], [8], [9].

The problem of performing color segmentation under varying lighting conditions is well known in the vision community under the term *color constancy* [10]. To our knowledge there is no approach using a color constancy algorithm for segmenting in real-time face and hands of a person that is subject to varying lighting conditions and that is observed from a moving camera. This is due to the fact that there is up to now no general solution to solve the color constancy problem in real-time [11]. Therefore, instead of *tolerating* lighting variations with a color constancy algorithm, current segmentation algorithms *adapt* to lighting variations (e.g. [12], [13], [14]).

It is only possible with algorithms capable of adapting to a changed appearance to segment human skin under lighting variations as they are encountered in unrestricted domains typical for mobile robots. Having an adaptive skin color segmentation method allows for efficient face detection and provides the basis for recognizing command gestures. While in the literature to our knowledge only the reduction of the image search space has been explored, we propose to use the result of the face detection process as context information for the adaptation step. If a success-

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

In Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN), pages 337-343, Berlin, Germany, September 2002. IEEE.

ful face detection has been carried out on a skin-colored region, this information provides a reliable cue for selecting the position and size of the image area that is used for updating the skin color model.

This paper describes our work on developing an adaptive skin color algorithm that explores the combined use of skin color segmentation and face detection in order to improve the overall performance. This algorithm forms part of our ongoing work to develop an advanced human-machine interface for a mobile robot. Tracking the user allows the robot to engage in a dialog, e.g. to learn new objects [15]. First attempts towards multi-modal human-robot interfaces are already demonstrating the potential of such integrated approaches for robot instruction (e.g. [8]).

The rest of the paper is organized as follows: In the next section we will review related work on skin color segmentation and face detection. The basic processing scheme of our approach is described in section III. Section IV introduces our adaptive method for skin color segmentation and section V covers our face detection approach. Results for the integrated approach are given in section VI. Following some remarks on our ongoing work towards a sophisticated human-machine interface in section VII the paper ends with a conclusion.

II. Related Work

As pointed out above, a basic prerequisite for human-machine interfaces is tracking of the human face which in turn needs skin color segmentation to speed up the detection process. As a mobile robot encounters situations with high variability in lighting conditions, an adaptive skin color segmentation method has to be used.

Probably the most famous adaptive image segmentation system is the Pfinder ("person finder") system [14] for tracking a single human in real-time. For this purpose a simple representation of the human body with head, torso, arms, hands, legs and feet is used. The color of each body part is modeled as Gaussian in YUV color space and its position as Gaussian in image coordinates. However, only a single, completely visible human wearing homogeneously colored clothes is allowed to enter the scene and the segmentation scheme relies on a nearly static background. The related LAFTER system [12] uses similar techniques to track and recognize the face of a single user sitting in front of a computer. For each color model (face, lips and interior of the mouth) a mixture of Gaussians is learned offline. For color representation the $r-g$ color space

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B}$$

is used as it is well suited for representing skin color over a wide range of lighting conditions [16]. Classification is performed by assigning each pixel to the class with the highest probability. Adaptation consists of updating each class with the assigned pixels through an iterative version of the EM-algorithm. As the distance between user and camera does not change much, the region with shape and size similar to the expected face is selected to represent the face.

A similar approach by Raja et al. [17] employs a single Gaussian for modeling skin color in $r-g$ color space to track the face of a single person in real-time. After initialization of the unimodal Gaussian the skin probability image is computed by calculating for each image pixel its probability. Subsequently, the face position is obtained by calculating the center of mass (COM) for the overall skin probability image. Here the mixture model is updated based on all pixels within a bounding box around the COM. In the example in [17] the active camera performs changes of pan, tilt, and zoom while the face of a human moving through an office undergoes large changes in illumination.

Recently, Soriano et al. [13] proposed an adaptive algorithm that tracks single faces on camera-equipped mobile phones. To reduce the computational load their algorithm employs a histogram representation for modeling the current face color distribution in $r - g$ color space. During initialization manually segmented skin regions are used to construct the skin histogram $S(r,g)$. A ratio histogram $R(r,g)$ that directly contains for every pixel color its skin color probability is obtained by dividing every bin in $S(r,g)$ by the corresponding bin in the histogram of the complete image $I(r,g)$. The ratio histogram allows to directly compute the skin probability image for a new input image. Similar to Raja et al. [17] the face is located by computing the COM of the skin probability image and the pixels in a bounding box are used for updating the histogram. To stabilize the updating process an empirically determined global skin color distribution is used for filtering out non-skin pixels. This so-called *skin locus* is described in more detail in section IV. The approach works well if a single face is positioned close to the camera, i.e. occupying a large area of the camera image.

For human-machine interfaces tracking of the user's face is indispensable. It provides information about the user's identity, state, and intent. A first step for any face processing system is to detect the locations of faces in the robot's camera image. However, face detection is a challenging task due to variations in scale and position within the image. In addition, it must be robust to different lighting conditions and facial expressions. A wide variety of techniques have been proposed, for example neural networks [18], deformable templates [19], skin color detection [20], or principle component analysis (PCA) [21]. For an overview the interested reader is referred to [22], [23].

Similar to our approach, there are face detection methods based on skin color with a subsequent verification step

[24], [25], [26]. First, skin colored regions are segmented from the image. Regions of elliptical shape are selected as face candidates. For verification, local features such as eyes and mouth are extracted from the regions. Since these approaches process still images only no adaption of the skin color models is performed.

In contrast to extracting local features for verification, we use an appearance-based approach. A popular approach is the *eigenface method*, operating on intensity information of images. Any graylevel image with a size of $n \times m$ pixel can be considered as a point in a nm-dimensional space. Faces lie in a subspace of the overall image space. Kirby and Sirovich demonstrated in [27] how PCA can be used to efficiently represent human faces. Later, Turk and Pentland [21] applied this technique to face detection. PCA finds the principle components of the distribution of the face images, which are called *eigenfaces*. They span a subspace (*face space*) representing possible face images. Any image with a size of $n{\times}m$ pixels can be reconstructed by a weighted sum of eigenfaces. Since the reconstruction is an approximation, a residual error ε emerges which is small for face images and large otherwise. Hence, ε can be used as a feature when distinguishing face from non-face images.

III. Architecture

Our processing scheme realizes a closed-loop color adaptation by using the results of the face detection process to select the image area that is used for adapting the color model. In order to allow fast processing only portions of the image are segmented with local skin color models that adapt to the object, i.e. face or hand. The prediction of the skin color model for segmenting the next image based on image data that is selected during a verification step, i.e. the face detection, bears similarities to other techniques that use measurements to correct a prediction, e.g. Kalman filtering. The basic processing loop consists of three steps that are executed for every new image (see Fig. 1):

1. Image segmentation with local skin color models around predicted object positions.
2. Face detection process starting at the center of skin-colored regions.
3. Adaptation of the skin color models based on image areas resulting from faces found.

In the initialization step an *object* is instantiated for every skin-colored region in the image. Assuming that only human hands or faces have skin color, an object represents either a hand or a face. While a successful face detection allows to assign the label *face* to a skin-colored region, we currently have no means to recognize whether a region belongs to a hand based on image data alone. This information could be inferred if a model of the human body and temporal information about the movements of the regions

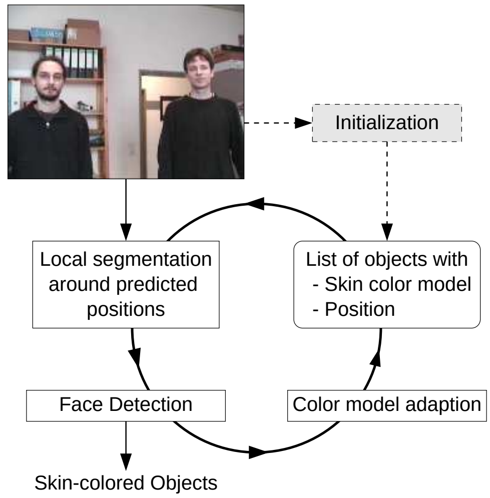


Fig. 1. Basic image processing loop for adaptive color segmentation with image areas for updating based on face detection results.

is used [9]. We will implement a similar method in the future, but for now we assign the label *non-face* to every region that was not successfully recognized as face.

The position of all objects, i.e. regions with *face* or *non-face* labels, is tracked to perform local image segmentation. In every new image a region segmentation is carried out only around the predicted positions of the objects currently in the object list. The object positions are updated with the actual positions of the segmented regions and objects that have disappeared are deleted from the object list. Since searching for newly appeared objects would require the computationally expensive processing of the complete image, there is currently no process to add new objects. At every time step a face detection is carried out for all tracked skin-colored regions to recognize whether a human is looking at the camera, i.e. the robot, and to detect faces of humans that have not faced the camera previously. If a face was recognized, an elliptical region approximating the face size is used for updating the skin color model. Regions that do not contain a face, e.g. because of a non-frontal view of a face, are not updated. The updated model is used in the next image for segmenting the image area associated with this model through the predicted position.

IV. Adaptive Skin Color Segmentation

Different from the approaches sketched in Section II our goal is the tracking of *several* skin-colored image regions. This is realized by incorporating context knowledge from a face detection algorithm (see Section V) and combining the techniques used by Raja et al. [17] and Soriano et al. [13] for performing adaptive skin-color segmentation. As the different skin-colored areas in an image may be subject to different lighting conditions, we model the skin-color distribution for each image region separately. The ratio histogram calculation cannot be extended easily to the case where multiple skin-colored image regions are present that have to be modeled individually. Using separate ratio histograms results in a poor modeling due to the small amount of skin pixels in the individual skin-colored image areas present in our domain. Therefore we use Gaussian models to represent individual skin color distributions. To keep the computational load small we use an unimodal Gaussian instead of a mixture of Gaussians as this has been shown to be sufficient for the special case of modeling a person's face [28]. The skin probability of a pixel $\mathbf{x} = (r, g)$ based on the Gaussian i is calculated from:

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)\sqrt{\det \mathbf{\Sigma}_i}} \exp\left\{ -\frac{1}{2}[\mathbf{x} - \mu_i]^T \mathbf{\Sigma}_i^{-1}[\mathbf{x} - \mu_i] \right\}$$

For initialization of the Gaussian parameters μ and Σ for all models we assume that all faces that have to be tracked are visible in the current image. As the face detection algorithm is based on grey level images only, skin color models for face regions are bootstrapped by performing a face detection on the complete image. Searching for faces in the complete image takes several seconds. However, this is tolerable since the search is only carried out during the initialization phase. By supplying an estimate of the face size, i.e. the distance of a person, the time necessary for face detection can be reduced significantly. If at least one face is found, an initial skin color model is now available. If several faces are found, an average model is calculated. Based on this initial Gaussian model the skin color probability is computed for every image pixel to obtain an initial skin probability image. Now a threshold is used to segment the image into skin and non-skin regions. Following a connected components analysis the initial object list is constructed by adding for every region found an entry that contains the region position and the initial Gaussian parameters.

After initialization, classification of the input image at time t is restricted to regions of interest (ROI) based on the positions of the skin-colored regions contained in the object list. For each object i its current Gaussian parameters are used to classify a region of interest $ROI(i)_{classify}$ that is located at the $COM(i)_{t-1}$ of the region i segmented in the previous time step. This skin probability area is segmented into a binary skin/non-skin image by applying a threshold. Subsequently, for each $ROI(i)_{classify}$ the region size, its surrounding polygon, and its new $COM(i)_t$ are calculated. This new COM is used for updating the tracking process and as starting point for the face detection.

Besides using the face detection result for the human-machine interface, this context knowledge allows to shape the image area used for updating the Gaussian model.

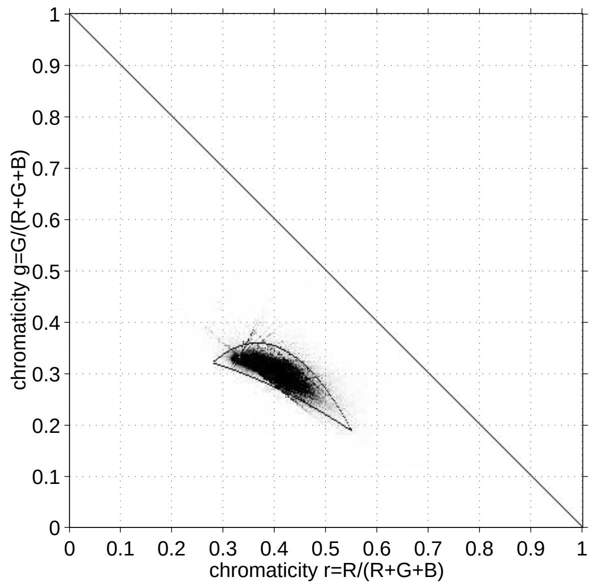


Fig. 2. The measured skin color distribution in $r - g$ color space with the two quadratic functions F_u and F_d.

While others have used simple rectangular areas (e.g. [17], [13]), we use an elliptical ROI_{update} if the region was classified as *face*. If no face was detected, no update is performed.

Similar to Soriano et al. [13] we employ an empirically determined global skin color distribution to restrict the updating process to skin-colored pixels. This is done by using hand-segmented image patches of human hands and faces obtained from several different people under many different lighting conditions typical for our indoor domain. Following a theoretical model it has been shown by Störring et al. [29] that the overall skin color distribution is a shell-shaped area in the $r - g$ color space that is called *skin locus*. For our Sony EVI-D31 camera the skin color distribution and the two quadratic functions to approximate the skin locus are shown in Fig. 2. The parameters of the two quadratic functions that enclose 95% of the pixels in the empirically determined skin color distribution are

$$\begin{aligned} A_u &= -5.05 & A_d &= -0.65 \\ b_u &= 3.71 & b_d &= 0.05 \\ c_u &= -0.32 & c_d &= 0.36 \end{aligned}$$

The decision whether a specific pixel $\mathbf{x} = (r, g)$ is skin-colored and can be used for model adaptation is now readily available by calculating the two quadratic functions based on the r value

$$F_i = A_i r^2 + b_i r + c_i \quad with \quad i \in \{u, d\}$$

and checking for the g value to lie between the two func-

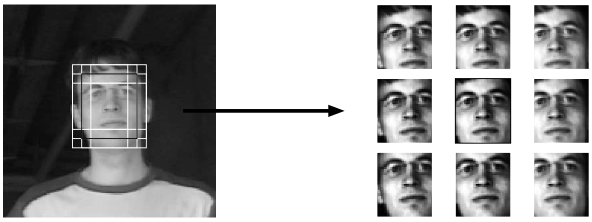


Fig. 3. Generation of eight variants with different positions (white) according to the initial selection (black).

tions:

$$\text{Pixel is skin} = \begin{cases} 1 & \text{, if } (g < F_u) \text{ \& } (g > F_d) \\ 0 & \text{, else} \end{cases}$$

All pixels within ROI_{update} that lie inside the skin locus are used to smoothly update the mean and covariance of the Gaussian model:

$$\begin{aligned} \vec{\mu}_{new} &= \gamma\, \vec{\mu}_{(ROI\_skinlocus)} + (1-\gamma)\, \vec{\mu}_{old} \\ \mathbf{\Sigma}_{new} &= \gamma\, \mathbf{\Sigma}_{(ROI\_skinlocus)} + (1-\gamma)\, \mathbf{\Sigma}_{old} \end{aligned}$$

Given operation with a frame rate of 3 Hz a learning rate of $\gamma = 0.6$ has been shown to provide good results for persons moving in a standard office domain. In general the learning rate should be chosen depending on the speed of lighting variations exhibited by a moving face. A compromise has to be found between a high value in order to allow for adaptation to lighting changes and a low learning rate to avoid adaptation to a temporary face appearance.

V. Face Detection

In the previous section we have shown how skin colored regions can be segmented and tracked in image sequences. Now we describe the verification step, which checks for every region whether it is a face or not. We use the eigenface method for face detection [21].

A prerequisite for the effective use of the eigenface method is to select a suitable set of sample images for PCA. Since eigenfaces should only represent variations originating from variances of faces of different persons, all other variations should be reduced. We exclude variations of the background by only selecting the central part of a face. Varying lighting conditions are compensated by preprocessing the graylevel images using histogram equalization.

Additionally, there are variations in position and size of the faces within the different sample images, e.g. the eyes are not always centered or at the same height. This results from manually segmenting the samples from larger images. In order to reduce this effect, for every manually segmented sample face f_i $(i = 1 \dots N)$ a set of K variants V_i is generated (notice: f_i itself is considered as a variant):

$$V_i = \{v_{i,1}, \dots, v_{i,K}\} \quad \text{with} \quad v_{i,1} := f_i$$

Each variant v_{ik} $(k>1)$ is segmented with slightly different position and scale according to the initial selection v_{i1}. Fig. 3 shows an example for generating eight new variants with different positions from an initial face segmented manually.

Any selection S consisting of one variant from each face serves as a set of training faces for PCA:

$$S = \{s_1, \dots, s_N\} \quad \text{with} \quad s_i \in V_i$$

To extract a good selection we propose an automatic algorithm that iteratively optimizes the set of training faces. The algorithm starts at $t = 0$ with an initial selection S_0, which consists of the manually segmented faces:

$$S_0 = \{s_{0,1}, \dots, s_{0,N}\} \quad \text{with} \quad s_{0,i} = v_{i,1}$$

In order to determine a new selection S_{t+1} based on S_t the function $NewSelection$ is defined:

$$\begin{aligned} NewSelection(S_t, r) &= S_{t+1} \\ &= \{s_{t+1,1}, \dots, s_{t+1,N}\} \text{ with} \end{aligned}$$

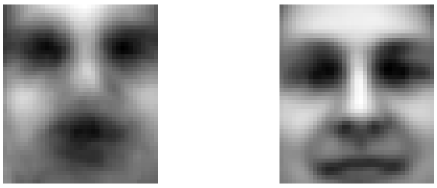
$$s_{t+1,i} = \begin{cases} \underset{v_{ij} \in V_i}{\operatorname{argmin}}\, \varepsilon\left(F(S_t \setminus \{s_{t,i}\}),\, v_{ij}\right) & \text{, if } (i = r) \\ s_{t,i} & \text{, else} \end{cases}$$

Thereby $F(\cdot)$ denotes the face space calculated from a given set of variants, and $\varepsilon(\cdot,\cdot)$ denotes the residual error resulting from reconstructing a face image with a given face space. The function $NewSelection$ replaces the rth selected variant $s_{t,r}$ while all other selected variants $s_{t,i}$ $(i \neq r)$ remain unchanged. For replacement all K variants $v_{rj} \in V_r$ belonging to the corresponding face f_r are reconstructed by a face space $F(S_t \setminus \{s_{t,r}\})$ computed from the remaining selected variants. The one which has minimum reconstruction error ε will become the new selected variant $s_{t+1,r}$.

The overall algorithm that iteratively replaces randomly chosen variants using $NewSelection$ until a stable state is reached is summarized by:

$$\begin{aligned} & S_0 \leftarrow \{s_{0,1}, \dots, s_{0,N}\} \quad \text{with} \quad s_{0,i} = v_{i,1} \\ & t \leftarrow 0 \\ & \textbf{repeat} \\ & \quad r \leftarrow rand(N) \\ & \quad S_{t+1} \leftarrow NewSelection(S_t, r) \\ & \textbf{until } (NewSelection(S_{t+1}, i) = S_{t+1}\ \forall i) \end{aligned}$$

The capability of this approach is demonstrated in Fig. 4. It shows the mean face images calculated from the initial selection (on the left), and the optimized selection (on the right). Notice the more distinct face structures extracted by our algorithm that result in a better recognition accuracy.



(a) (b)

Fig. 4. Mean face images calculated from (a) the initial selection, and (b) the optimized selection.

After the eigenfaces have been constructed from the optimized set of face samples, new images can be classified as *face* or *non-face* depending on the reconstruction error.

In order to decide whether a segmented region of skin color originates from a face, a sub-image at the position of the region has to be extracted and classified with the eigenface method. In case of a face, the sub-image not necessarily coincides with the face due to inaccuracy of segmentation. Therefore, the area at the region has to be scanned at different positions and with varying scalings. Since this is computational expensive, we propose an intelligent scanning process which reduces the search space.

The center of the initial sub-image (x, y) coincides with the COM of the skin colored region. At this and the two neighboring positions $(x+1, y)$ and $(x, y+1)$ the corresponding reconstruction errors for the extracted sub-images are computed. The next position of the scanning process is chosen according to *steepest gradient descent*. This process stops if a face is detected or a local minimum is reached. In the latter case the process continues with sub-images of a new size ($\pm 7.5\%$).

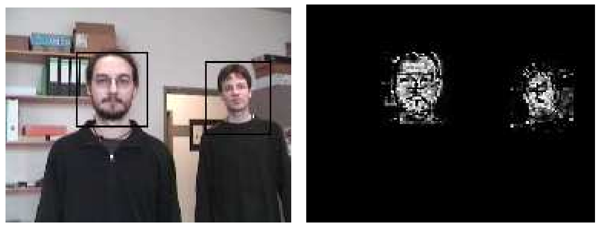
So far the set of sample images consists of faces in frontal view, so that only faces directed towards the camera can be detected. This can be extended to multi-view face detection using additional detectors, each of them for one specific view [30].

VI. Results

The color images are obtained from a Sony EVI-D31 pan-tilt camera mounted on top of an ActivMedia PeopleBot robot (see Fig. 5). Since the standard onboard PC is used for control of the robotic platform, we use an additional onboard PC (500 MHz Pentium III) mounted in tall extension for image acquisition and image processing.

Our face tracking system is implemented in C/C++ and is based on an existing image processing framework [31]. Currently we use an image size of 192×144 pixels which results in a frame-rate of 15 Hz if the face detection algorithm is not used. With the face detection the frame-rate drops to approximately 3 Hz for scenes with 2 skin-colored regions. As every skin-colored region is searched for a face, the frame rate heavily depends on the number of tracked regions. Figure 6 shows an example classification result for a typical office scene.

Fig. 5. Example interaction between robot and human.



(a) Input image with ROI (b) Classification result

(c) Segmented regions (d) Face ellipse for Update

Fig. 6. Example for adaptive color segmentation with a) the ROI area used for classification, b) the skin probability image, c) the region polygons obtained with a classification threshold and d) an update area (black ellipse) from face detection.

VII. Ongoing work

Currently we are integrating the presented face tracking in a larger framework to teach a robot to supply water to selected indoor plants. This is a practical and relative simple problem, which provides us with a wide scenario for human-machine-interaction. Watering plants is also investigated in the PlantCare project [32], where the focus is on ubiquitous computing and the plants are equipped with wireless sensors. Especially for homecare robots (e.g. [33]) plant watering is an important task.

While in the PlantCare project human-machine-interaction is minimized, it is the main focus in our scenario. Our goal is to enable a robot to follow a person going to an arbitrary plant. Then the robot should track a pointing gesture to the plant location and interactively learn the visual appearance of the plant [15]. The position of the plant and additional information given by the user in natural speech will be stored in order to recover the position of the plant at a later point of time and to supply it with a specified amount of water.

VIII. Conclusion

The presented adaptive skin color segmentation allows to segment skin color areas that undergo changes in lighting conditions. The additional information from the face detection process is incorporated to improve the adaptation step by using elliptical image areas at detected face positions for updating the skin color model. With the position information from the pan-tilt camera we can calculate a rough position of the segmented objects in robot coordinates to allow for tracking. The presented system forms the basis of ongoing work to develop more sophisticated human-robot interfaces capable of dealing with unrestricted environments.

References

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

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

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

[4] V. Graefe and R. Bischoff. Three examples of learning robots. In *Proc. Int. Conf. on Control, Automation and System*, 2001.

[5] S. Fischer and K. U. Simmer. Beamforming microphone arrays for speech acquisition in noisy environments. *Speech Communication*, 20(3–4):215–227, 1996.

[6] M. S. Brandstein and H. F. Silverman. A practical methodology for speech source localization with microphone arrays. *Computer Speech & Language*, 11(2):91–126, 1997.

[7] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

[8] S. Ghidary, M. Hattori, S. Tadokoro, and T. Takamori. Multi-modal human robot interaction for map generation. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2246–2251, 2001.

[9] C. Wengert, T. Fong, S. Grange, and C. Baur. Human-oriented tracking for human-robot interaction. Int. Conf. on Multimodal Interfaces, 2002. To appear.

[10] D. Forsyth. A novel algorithm for color constancy. *Intl. J. Computer Vision*, 5:5–36, 1990.

[11] B. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? *Lecture Notes in Computer Science, ECCV'98*, 1406:445–459, 1998.

[12] N. Oliver, A. Pentland, and F. Berard. Lafter: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33:1369–1382, 2000.

[13] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 839–842, 2000.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Vision*, 19(7):780–785, 1997.

[15] F. Lömker and G. Sagerer. A multimodal system for object learning. In *Pattern Recognition, Proc. of 24rd DAGM Symposium*, Lecture Notes in Computer Science, Zürich, 2002. Springer. To appear.

[16] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. *Lecture Notes in Computer Science*, 1352:687–694, 1998.

[17] Y. Raja, S. J. McKenna, and S. Gong. Colour model selection and adaptation in dynamic scenes. In *ECCV*, pages 460–474, 1998.

[18] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[19] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *Int. Journal of Computer Vision*, 8(2):99–111, 1992.

[20] J. Yang and A. Waibel. A real-time face tracker. In *Proc. of WACV'96*, 1996.

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

[22] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[23] E. Hjelmås and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.

[24] J. Sobottka and I. Pittas. Segmentation and tracking of faces in color images. In *Proc. IEEE Int. Conf. on Automatic Face & Gesture Recognition*, pages 236– 241, 1996.

[25] E. Saber and A. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 19(8):669–680, 1998.

[26] M.-H. Yang and N. Ahuja. Detecting human faces in color images. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 127–130, 1998.

[27] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[28] Y. Raja, S. McKenna, and S. Gong. Segmentation and tracking using colour mixture models. In *Proc. 3rd Asian Conf. on Computer Vision*, pages 607–614, 1998.

[29] M. Störring, H. J. Andersen, and E. Granum. Physics-based modelling of human skin colour under mixed illuminants. *Robotics and Autonomous Systems*, 35(3–4):131–142, 2001.

[30] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *Proc. IEEE Int. Conf. on Automatic Face & Gesture Recognition*, pages 300–305, 2000.

[31] J. Fritsch, F. Lömker, M. Wienecke, and G. Sagerer. Detecting assembly actions by scene observation. In *Proc. Int. Conf. on Image Processing*, volume I, pages 212–215. IEEE, 2000.

[32] A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S. B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. Making sensor networks practical with robots. Int. Conf. on Pervasive Computing, 2002. To appear.

[33] C. Schaeffer and T. May. Care-o-bot: A system for assisting elderly or disabled persons in home environments. In *Proc. European Conf. for the Advancement of Assistive Technology*, pages 340–345, 1999.