Pointing Arm-Posture Recognizing Using Stereo Vision System[*]

BOGDAN KWOLEK

Rzeszów University of Technology, Computer and Control Engineering Chair
W. Pola 2, 35-959 Rzeszów
e-mail:bkwolek@prz.rzeszow.pl

Abstract: This paper describes an approach to real-time detecting static pointing arm-postures which can be used to send position instructions to a mobile robot. This takes place by analyzing the color image and utilizing the range image of the scene obtained by a stereo vision system. First the algorithm locates the person's face and afterwards the arm-postures are recognized. The start as well as the end of the pointing command have been signaled by suitable static arm-postures of an instructor who stands in front of the robot. The system has been tested in tasks consisting in selecting the targets on the laboratory floor. All image processing is realized on-board the robot.

1. Introduction

Arm-postures can be a very useful and compact way to convey geometrical information to the robot [10]. One of the most expressively significant arm-postures is pointing towards an object [8], [18]. Having a robot with a recognition interface, the user can point to the desired spot on the floor, rather than input the location to where the robot should move. For non-expert users it is easier to point to an object than employ the verbal description of its coordinates.

[*] This work has been supported by the Polish Committee for Scientific Research (KBN) within the project 7T11C03420

The aim of this work was to build a system that makes possible realization of experiments which consist in determining which object from among those lying on the floor and being in the view of the camera the user has pointed to, giving commands in front of the autonomous mobile robot. The recognition of such commands has been realized on the basis of template matching techniques [17]. The centers of correlated sub-images with the manually prepared in advance templates have been located in the area which location had been determined on the basis of eyes position. The determination of eyes position has in term been preceded by a searching within skin-like regions for symmetrical face candidates as well as candidates of face which match suitable eyes-template. The final verification of face presence has been realized using the well known eigenfaces approach [20]. The information about the relative distance of the user's face to the camera has allowed us to avoid search over scale parameter in the matching procedure used for arm-posture recognition. The determination of floor spot which was pointed to by user and of the mentioned above scale ratio as well as the selection of the eyes-template size have been realized on the basis of stereo vision.

The contribution of this work relies on the employment of several visual cues and in particular stereo with the aim to detect and to localize the face and then to recognize in real-time static pointing arm-postures. By using the stereo system the positions of the person's face and hand in the scene are determined and used then by module responsible for determination of the direction of pointing. Since false-positives (the robot recognizes the command which was not given by the user) are in human-machine communication less desirable than false-negatives (the robot fails to recognize the hand-action), our approach rests on reliable face detection and localization.

Human face detection plays an important role in human interface. Some of the best techniques for face detection are still too computationally expensive and cannot match the real-time constraint using the personal computers that we have today. The eigenface approach is one of the few techniques that can fulfill the real-time requirements. However, most eigenface-based systems work with the assumption that the location of a face within image is known [25]. The use of pattern matching with the patterns being prepared manually is reported in work [3]. Since the face size and position were not known a priori the searching has been realized on all possible sizes and locations of the head. The system reported in [3] takes 2 seconds on a SGI Indigo 2. A system for real-time face detection and tracking on the basis of pan-tilt-zoom controllable camera has been presented in [23]. The face detection takes about 0.2 seconds on a Pentium 266 MHz. Detected skin pixels are grouped into regions and then regarded as faces if dark regions of eyes, eyebrows and mouth are detected within these regions. Sobotkka at al. [16] showed that human skin color for all races are clustered in normalized RGB space. In work [14] shape symmetry is used to verify whether it is correct face region or not. In work [22] a robot looks for dark facial regions within face candidates obtained on the basis of color. Distance to the tracked color is roughly determined by measuring the area covered by the color in the image. A binary matching technique is used to detect the face in area located above of the rectangular boundary of the detected color shirt. The sizes of sub-images are scaled according to the robot to person distance provided by the sonar sensors. In work [21] a fast and adaptive algorithm for tracking and following a person by robot-mounted camera has been described. After locating the person in front of the robot an initial probabilistic models of the person’s shirt and face colors are created. The system uses window of fixed size to adapt the face and shirt color models over time and it is essential that these windows can only contain face and shirt colors. In particularly, the distance between the robot and the person must be approximately fixed. Two alternative methods for pose analysis are used: neural networks and template matching. The Viterbi algorithm is utilized to recognize motion gestures. The Perseus system [8] is capable of finding the object pointed to by a person. The system assumes that people is only moving object in the scene. Perseus uses independent types of information e.g. feature maps and disparity feature maps. The distance between the robot and person may vary. In other system [15] a behavior for a person following with an active camera mounted on robot has been presented. In that system the head of person is located using skin color detection in the HSV color space and color thresholding techniques.

The organization of the paper is as follows. Color image processing is discussed in section 2. In section 3 the algorithm for face presence detection and face localization is described. Arm-posture recognition is presented in section 4. In section 5 experimental results are briefly overviewed. Finally, in section 6 a conclusion follows.

2. Color Image Processing

Color image segmentation can be useful in many applications. On the basis of segmentation results, it is possible to identify regions of interest and desirable objects. The problem of segmentation is not easy because of image texture, shadows, etc. If an image contains only homogeneous color regions, clustering methods are sufficient to handle the problem. In practice, natural scenes are rich both in color and texture as well as they are under varying lighting conditions. Therefore our approach uses color only for coarse extraction of regions of interest and additionally utilizes the range information. In particular, the range information allows us to reduce search over scale or select appropriate template size in the correlation procedure. Detection on the basis of matching is an expensive operation, involving search over the input image and over the scale during matching with the model. To reduce the working area in the matching procedure utilized in arm-posture recognition we have chosen the eyes location as the datum point. The search for eyes has been preceded by skin regions detection.

Face color as a feature for tracking people has been used for example in work [24]. The advantage of such approach relies on the speed at which present low-cost personal computers can extract the object of interest from color images. That aspect is particularly important considering on the one hand the limited computational power of on-board computer of the mobile robot and on the other hand the necessity of work at rate of around 10 Hz.

The most common space used in computer displays and CCD cameras is the RGB color space. Since the brightness can change rapidly when robot moves, most of the known color segmentation algorithms operate using chromatic colors. The authors of work [1] came to the conclusion that the best space for skin color representation is the normalized color space. The two-dimensional intensity-normalized chromaticity color space is relatively robust to the change of the illumination. But an obvious shortcoming is that the pure colors are unstable and meaningless when R+G+B is small [12]. Therefore following a suggestion from work [19] we have only employed the chromatic areas of image in further color processing.

The skin color distribution in normalized space of colors can be represented by a Gaussian model [24] $M = (\mu_r,\ \mu_g,\ \Sigma)$, where $\mu_r,\ \mu_g$ are the means and Σ is the covariance matrix. The parameters of Gaussian model have been obtained on the basis a set training images. Each image with user outlined regions of interest has the color space transformation. The aim of such an operation was to obtain a representative sample of typical skin colors of objects and thus it was not essential to outline the whole face. One attraction of bimodal normal distribution is that it can be used to generalize about small amounts of training data. To extract the face candidates, each pixel was examined by Gaussian model. The better the pixel matches the color model, the higher the probability and response of such a color filter is, see fig. 1b.

Filtering on the basis of the mask whose weights correspond to the binomial coefficients [5] was the first operation realized on acquired images. Then the filtered RGB values were transformed into probability ones. To shorten the execution time the lookup table for direct conversion from RGB to probability was utilized.

A pixel was identified as a candidate of the face if the corresponding probability was above a threshold. The threshold has been set up to a low value. Because we have been interested in coarse extraction of face candidates, the choice of threshold did not have a considerable influence on obtained results. After an image was thresholded, morphological closing operator (dilate followed by erode) [7] was performed. The aim of such an operation was to fill small holes and to smooth border.

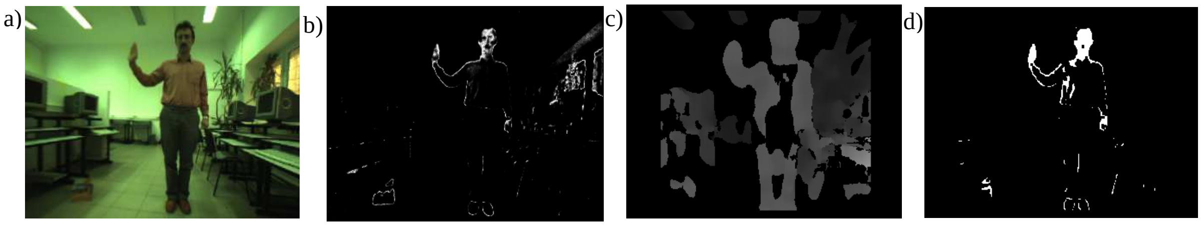

Figure 1. Face candidates extraction. Raw color image (a). Probability image (b). Depth image (c). Closed image (d)

Our system utilizes the depth information provided by the commercial SRI Megapixel Stereo Head [9]. Corresponding pixels between images are detected using the area-correlation method where the correlation is used to obtain the most correspondences between small patches [9]. We have set up the number of disparities to 32 and thus worked with a minimal horopter distance of 96 cm and a maximum range resolution of 7.5 mm. Stereo depth information that was utilized at this stage can be found in small area correspondences between image pairs and therefore it gives poor results in regions of little texture, see fig. 1c. However, the depth map covering a face region is usually dense because the human face is rich in details. The person performing hand-actions has stayed about 1.5 m in front of the robot. Taking the depth image, we have removed from closed image the pixels representing a distance to the camera greater than 2.5 m, see fig. 1d. The image prepared in such a way was then labeled. The objective of the realization of this operation was to determine distance of each individual scene object to the camera.

The connected component analysis was utilized to gather neighborhood pixels into larger regions. Connected component labeling scans an image, pixel-by-pixel, from top to bottom and from left to right in order to extract pixels and regions which share the appropriate set of intensity values. Each object receives a unique label which can be used to distinguish regions during further processing. In our approach the connected components algorithm works on the binary image and is based on region identification in 8-connectivity [17]. At each point the algorithm examines pixel *p* from binary image and if that pixel is set to foreground, it checks its four neighbors from labeled image which have already been processed (i.e. the neighbor to the left of actually analyzed pixel, above it, upper left and right). Having obtained values of pixels, one of the following can occur:

if all four neighbors are 0, assign a new label to pixel *p*, else

if only one neighbor has a label, assign its label to pixel *p*, else

if one or more of the neighbors have two different labels, assign to *p* a label which is smaller and update the glue table for equivalencies.

Each pixel is replaced in the second pass by one of the unique labels which had been assigned to each region and stored in the glue table in the previous pass. The labels obtained in such a way were then used within the depth image to determine an average range of each labeled region.

3. Face Presence Detection and Face Localization

The matching procedure which was used for arm-posture recognition has been preceded by face localization module. The reliable face location has allowed us to decrease the number of sub-images with expected arm-postures because the operating area of matching procedure was bounded to the region that surely contained the arm and body of the user. Thanks to such a datum point we had to re-center the arm-templates in relative small surrounding and thus the recognition was realized fast and reliable. Stereo range gives information how distant objects are and thus for objects of known size such as people who take place in our experiments we could set the image scale for arm-postures matching.

Face detection methods can be divided into four groups [25]. In knowledge-based model techniques, the model consists of features and their relation between each other is derived from our knowledge of human faces. The relations of features are expressed as the sets of rules and the verification of hypotheses on the basis of a decision tree is integrated within a search for a face. The methods based on feature-invariant approach also search for facial features. In this approach the face candidates are extracted by maximizing search criteria. The set of face features is assumed to be invariant regardless of the face viewpoint, orientation. These features can be geometric or skin color (texture) based. The approach which is based on template matching maximizes a correlation function of a human face template over the whole image. Sample image window with a face appearance is sometimes normalized and scaled. The template is correlated and in the point where the correlation has the largest value the hypothesis is then formulated. The appearance-based methods operate in a multidimensional feature space. In the methods that are based on eigenfaces the feature space is reduced using principal components. The scaled query windows of the input image are projected into classification subspace and the closer the distance to the projected training image is, the greater the probability that such window corresponds to the trained human face is. Our approach to face detection relies on mixture of the last three techniques. After face detection we know exact three-dimensional position of face with respect to the camera.

In face authentication system based on eigenfaces approach, a symmetry measurement as a preprocessing step helps in discarding face candidates regions that do not need to be considered reducing in the consequence the detection time. The first stage of our algorithm for face localization operates in gray images and it searches for symmetrical face candidates. The vertical axis of symmetry has only been considered. A symmetry measurement based on the correlation coefficient has been used to locate the center lines of face candidates. The non-zero pixels of closed image have only been considered as centers of windows in which the symmetry has been measured. Our measurement of symmetry has been based on the correlation between a region covered by the left half of the rectangular window and its reflection with respect to its vertical center line:

$$sym[row][col] = \frac{1}{\sigma^2} \sum_{i=-N}^{N} \sum_{j=1}^{K} \left(I[row+i][col-j] - \bar{I}\right) * \left(I[row+i][col+j] - \bar{I}\right) \tag{1}$$

where $\bar{I}$ is the mean value over the area being correlated and σ is the standard deviation over the area being correlated. Thanks to the being in disposal information about distance of the examined candidate of face to the camera, the reflected symmetry has been checked in windows about sizes needed to cover only the face region. The sizes N, K of the rectangular window have been taken proportional to bounding box of the face at specific location. The correlation coefficient of value near one means high symmetry while values minus one stand for anti-symmetry. The location of the axis of symmetry has been obtained by searching for the symmetry peeks of correlation coefficients within each row segment of the identical label, see fig. 2.

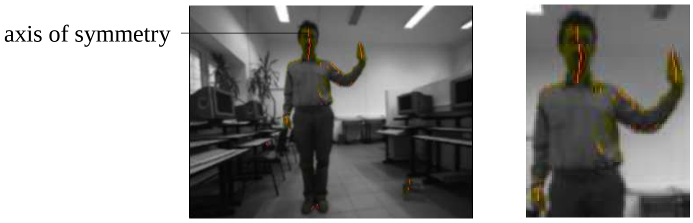

Figure 2. Symmetrical face candidates extraction

Next, taking again into consideration the range information, the eyes-template of appropriate size has only been applied in the centers of symmetry detected at previous stage. Among the miscellaneous facial features eyes are the most suitable for estimation of head pose. Most approaches to eyes localization are template based [2]. The template matching is a pattern matching technique that allows us to search for a pattern in an image. A search is realized by assigning a match score to each pixel, based on how closely the model matches the region around that pixel over which template is defined. The basic principle of template matching involves detecting objects in an image where a high degree of correlation exists between a specified template and regions of interest in the image. The normalized cross-correlation between eyes candidate and template can be computed as [6]:

$$cor[row][col] = \frac{\frac{1}{(2N+1)(2K+1)}\left\{\sum_{i=-N}^{N}\sum_{j=-K}^{K}\left(I[row+i][col+j]*M[N+i][K+j]\right)\right\} - \bar{I}*\overline{M}}{\sigma_I \sigma_M} \qquad (2)$$

where $(2N+1)*(2K+1)$ is the number of pixels in the model, I is the image against which the pattern is being matched , M is the model, $\bar{I}$ is the mean value over the area being correlated, $\overline{M}$ is the mean value of the template, σ_I is the standard deviation of I in the region under template and σ_M is the standard deviation of the template. Such a correlation-based technique assumes that all pixels of a facial image are equally important.

Templates that model human eyes have been formed based on real face images of different persons. To speed-up the matching process we have utilized additionally the binary templates which had been constructed on the basis of gray templates. The arrangement of elements in sample binary template is shown in fig. 3. The gray values in applied template symbolize the regions not considered in matching process, whereas the white and black areas symbolize regions which should be above or under the mean value, respectively. The template was compared, pixel by pixel, to gray pixel blocks recentered in the centers of symmetry. By using the binary template the pixels marked as the gray values have been omitted in the correlation. Next, in order to determine whether the correlation is high enough to indicate the match, we compared the others pixels of the gray image that were inside the template against the mean value of the region with the relation to the binary values indicated in the corresponding entry of the binary template, and then according to the results of the comparison

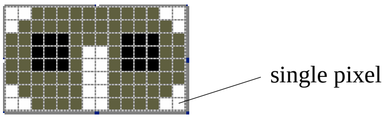

Figure 3. Matching template used in searching for eyes candidates

updated the correlation values in (2) or not. By using a number of thresholds some pixels with intensities compared to mean value have not been included in the correlation. A pixel block was considered as eyes if the matching score was greater than an acceptance level. The locations detected in such a way were then used by the face presence detection procedure that was based on the eigenfaces algorithm.

The aim of the usage of the eigenfaces was to reduce false-positive errors of face localization. The arm-postures have been recognized by matching procedure on binary images. In the case of false-positive localization of face, the localization of matching template is incorrect, and thus the output value of matching process can reflect commands which were not given by the user. Therefore the reliable face detection and localization play an important role in our algorithm of given hand-postures recognition. The reliable face detection can be very useful in several robotic tasks [4].

The eigenfaces are normalized eigenvectors which are the principal components of face space and they reflect the statistical properties of facial appearance. The eigenfaces algorithm operates on gray images and collection of training faces. We have prepared a collection of training faces $\Gamma_1, \Gamma_2, ...\Gamma_M$ that consists of $M = 64$ images. While preprocessing, the average image for this collection was computed as $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$. Next, this image was subtracted from each training image and a new set of vectors $\Phi_i = \Gamma_i - \Psi$ has been created. However, since the number of the face images is much smaller than the dimension of face space, there only exists $M-1$ nontrivial eigenvectors with the remaining ones associated with zero or negative eigenvalues. A common theorem in linear algebra states that the eigenvalues of AA^T and $A^T A$ are the same. Furthermore, the eigenvectors of AA^T are the same as the eigenvectors of $A^T A$ multiplied by the matrix A and normalized. If we consider V_i to be the eigenvectors of matrix $A^T A$, then $(A^T A)\, V_i = \lambda_i V_i$, where λ_i are the eigenvalues, $A(A^T A)\, V_i = A(\lambda_i V_i)$, which means $(AA^T)(AV_i) = \lambda_i (AV_i)$ and AV_i then are the eigenvectors of AA^T. The matrix $A = [\Phi_1, \Phi_2, ..., \Phi_M]$ was used to create the covariance matrix $A^T A$. The eigenvalues V_i and eigenvectors λ_i of such a covariance matrix were determined. The eigenvectors of

AA^T have been computed as $U_i = AV_i = \sum_{k=1}^{M} v_k^i \Phi_k$, where $i = 1,...,M-1$ and v_k^i is the *k*th element of V_i . The first ten eigenfaces related to our training collection are shown in fig. 4.

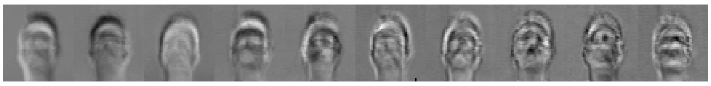

Figure 4. The first ten eigenfaces computed from our training collection

Since $U_i^T U_i = 1$, then $U_i^T AA^T U_i = \lambda_i$, and $\lambda_i = \frac{1}{M} U_i^T \sum_{k=1}^{M} \Phi_k \Phi_k^T U_i = \frac{1}{M} \sum_{k=1}^{M} \left(U_i^T \Phi_k\right)^T \left(U_i^T \Phi_k\right)$. Thus eigenvalue *i* represents the variance of the representative facial image along the axis described by eigenvector *i*. The first $M' = 16$ normalized eigenvectors, sorted by decreasing eigenvalue represent subspace in which classification at run-time phase was performed.

Prior to the run-time phase the set of reference images was read and projected into the classification subspace. A face image Γ can be projected onto eigenvectors using $\omega_k = U_k^T (\Gamma - \Psi)$, where $k = 1,...,M'$. The weights form a vector $\Omega^T = [\omega_1, \omega_2,..., \omega_{M'}]$ which describes the contribution of each eigenspace in representing the input image. In run-time the L1 distance between these vectors and the projected (and centered) image onto the subspace was computed to determine the closest match. The size of the sub-image which has to be cropped was scaled according to the distance from the robot to the person. The location of the sub-image was determined on the basis of eyes location.

The eigenfaces method is more robust than other template matching techniques that are based on detection of visible local features as well as distances between them. The algorithms based on eigenfaces recognize only a single face appearance that has been taken from a narrow angle, most often in the frontal view. Thanks to the obtained in advance localization of face candidates, the presence of a vertical and frontal-view face in the scene was verified in a very fast manner.

4. Arm-Posture Recognition

The arm-postures that have been used by our system can be divided into two categories, i.e. signaling the start as well as the end of the pointing command and the proper pointing command. The recognition of the first category command relies on the matching templates, see fig. 5. b, c. The area which was not considered in the matching process was marked with suitable gray value. The body of the user and its surrounding background were marked by white and black values, respectively. The arm and its surrounding background received gray values that are close to white and black values, respectively, see fig. 5. b, c. The five weights inversely proportional to the occupied areas were used during the matching process with the mentioned templates.

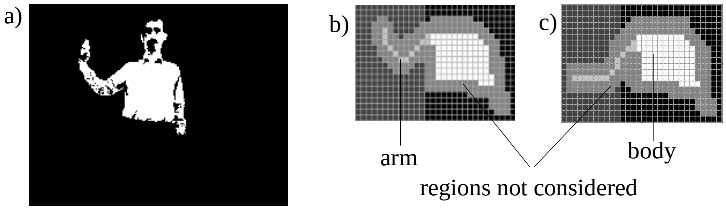

Figure 5. Sample binary image used in arm-posture recognition (a). Templates (b), (c)

The region growing is the process of grouping adjacent pixels or a collection of pixels which share similar attributes into larger regions [7]. The algorithm starts with a number of seed pixels and then from those it expands the regions by adding unassigned adjacent pixels that satisfy a desired criterion of similarity with the seed pixels. This algorithm was applied in our approach to extract the shirt of the user. The seed region was located in the area below the face where a depth compared with the depth of the face was detected. The following homogeneity criteria were used in a two-stage region growing

$$\sqrt{(I-\bar{I})^2+S^2+\bar{S}^2-2S\bar{S}\cos(H-\bar{H})}\leq d_1, \qquad |D-\bar{D}|\leq d_2 \tag{3}$$

where *H*, *S*, *I* (hue, saturation, intensity) are current values of color components of tested pixel, *D* is depth, $\bar{H}$, $\bar{S}$, $\bar{I}$, $\bar{D}$ are mean values related to the seed region, d_1 and d_2 are threshold values. The HSI criterion considers all three color components and takes into account the cylindrical nature of HS components. For reasons of angular value of hue the values belonging to the seed region have been converted into Cartesian coordinates in the following manner:

$$x_i=\cos(H), \quad y_i=\sin(H) \tag{4}$$

The average values $\bar{x}=\frac{1}{N}\sum_{i=0}^{N-1}x_i$ and $\bar{y}=\frac{1}{N}\sum_{i=0}^{N-1}y_i$ of the seed region were used in the arctan function for computing the mean $\bar{H}$ value. After the first stage of the region growing was finished, the average values $\overline{H}, \overline{S}, \overline{I}$ of the detected region were extracted and then in the second stage the HSI criterion with obtained in this manner new seed region and reduced to half d_1 value were used. The binary image which corresponds to the extracted regions of interest is presented in fig. 5a. The binary sub-image containing the user's body was scaled before matching according to the robot to the person distance. The only disadvantage of the HSI color space is the costly conversion from RGB color space. We handled this problem by using lookup tables.

Once the start command had been executed, the pointing arm-posture was watched for. During the pointing phase the arm remains stationary for a short time. Once the face position was known, the procedure responsible for determination of direction of pointing looked for the person's hand, which were assumed to be located to the right of the face. As with the face detection, the position of the hand was determined on the basis of labeled as well as depth images. Next, the consistency checks have been realized to verify that face-hand relationship which can be characterized by distances, angles, etc., is physically possible. The distances of the face and the hand to the camera that had been obtained from the depth image as well as positions of them allowed us to obtain direction of pointing. Then a line from the person‘s face center to the pointing hand was found and a cone was centered around it, starting at the hand. We have assumed that between the start command and the pointing command the user occupies approximately the same position. Taking the above into consideration, we obtained the face center on the basis of the skin-color region which was situated near the detected one during the execution of the start command, see fig. 6.

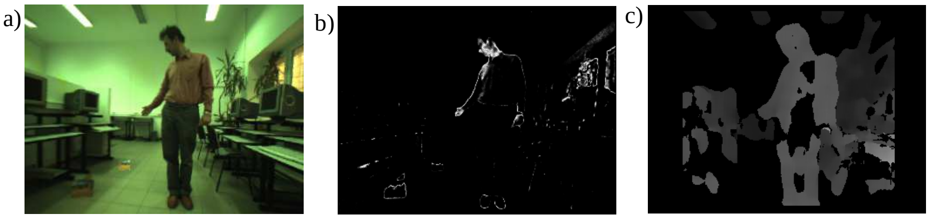

Figure 6. Pointing arm-posture recognition. The raw input image (a). Probability image (b). Depth image (c)

5. Experimental Results

We have presented an algorithm to recognize the pointing arm-posture which was implemented in system [11] and tested in several experiments. We have tested the algorithm on three people. The spatial resolution of pointing is about 20 cm and when the targets are at least 20 cm apart we can correctly tell a two targets apart in over 90% cases. The algorithm runs at 320x240 image resolution at frame rates of 6-10 Hz on 850 MHz Pentium III laptop which was installed on Pioneer 2 DX mobile robot [13].

6. Conclusion

We have presented a method to recognize the pointing arm-posture which can be used in tasks including robot programming by demonstration. The recognition has been performed on the basis of the information obtained from stereo cameras and the analysis of color as well as gray images. The combination of low-level image processing algorithms and of generic image features has allowed us to recognize the pointing command fast and robustly. One of the individual particularities of the presented method that makes it reliable is that it utilizes the information about the presence of the user's face. The depth map covering the face region is usually dense and this together with skin-color and symmetry information as well as eyes-template assorted with the depth has allowed us to apply the eigenfaces method and to detect the presence of the vertical and frontal-view faces in the scene very reliably. The face detection method consists of four significant stages: skin-like regions detection, labeling, searching for symmetrical face candidates and template matching, and the eigenfaces method. First, the color image is converted into binary image of skin-likelihood regions. The binary image is afterwards morphologically processed in order to prepare it more suitable for the next stage of image processing, that is labeling. The objective of labeling is to determine the distance of each individual object to the camera. The search for symmetry is applied to all the skin candidate regions in the binary image with the purpose of avoiding correlating non-frontal views with frontal eyes-templates. The template matching stage generally outputs a single position of eyes. Finally, the eigenfaces technique is used to decide whether the locations supplied by template matching stage belong in fact to face class, and it attempts to reject false detection that may have passed through the template matching or the symmetry search. The face detection method is fast and has a considerable degree of tolerance to luminance, color and face expression. By discarding the two/three most significant principal components, the performance of the face detection in the presence of lighting variation has been improved.

REFERENCES

[1] L. M. Bergasa, M. Mazo, A. Gardel, M. A. Sotelo, L. Boquete, Unsupervised and adaptive Gaussian skin-color model, Image and Vision Computing, 18, 2000, 987-1003.

[2] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Trans. on PAMI, vol. 15, no. 10, 1993, 1042-1052.

[3] Q. Chen, H. Wu, M. Yachida, Face detection by fuzzy pattern matching, In Proc. 5th IEEE Int. Conf. Comput. Vision, Boston, 1995, 591-596.

[4] S. Feyrer, A. Zell, Detection, tracking, and pursuit of humans with an autonomous mobile robot, In Proc. Int. Conf. on Intell. Robots and Syst., Kyongju Korea, 1999, 864-869.

[5] J. M. Gauch, Noise removal and contrast enhancement, In: S. J. Sangwine, R. E. N. Horne (eds.): The Colour Image Processing Handbook, Chapman & Hall, London, 1998.

[6] Gonzalez R. C., Digital image processing, Addison-Wesley Publ., London, 1977.

[7] B. Jähne, Digitale Bildverarbeitung, Springer-Verlag, Berlin Heidelberg, 1997.

[8] R. E. Kahn, M. J. Swain, P. N. Prokopowicz, R. J. Firby, Gesture recognition using the Perseus architecture, In Proc. of the IEEE Conf. on Computer Vision and Pattern Rec., San Francisco CA, 1996, 734-741.

[9] K. Konolige, Small Vision System: Hardware and implementation, In Proc. Int. Symposium on Robotics Research, Hayama Japan, 1997, 111-116.

[10] D. Kortenkamp, E. Huber, R. P. Bonasso, Recognizing gestures on a mobile robot, In Proc. of AAAI, 1996, 915-921.

[11] B. Kwolek, Person following, obstacle detection and collision-free path planning on autonomous mobile robot with color monocular vision, In Proc. of the 8th IEEE Int. Conf. Methods and Models in Automation and Robotics, Szczecin, 2002, vol. II, 971-978.

[12] Y. I. Ohta, T. Kanade, T. Sakai, Color information for region segmentation, Computer Graphics and Image Processing, 13, 1980, 222-241.

[13] Pioneer 2 mobile robots - Operations Manual, ActivMedia Robotics, LLC, 2001.

[14] E. Saber, A. M. Tekalp, Frontal-view face detection and facial feature extraction using color, shape symmetry based cost functions, Pattern Recognition Letters, vol. 19, 1998, 669-680.

[15] H. Sidenbladh, D. Kragić, H. I. Christensen, A person following behaviour for a mobile robot, In Proc. of the IEEE Int. Conf. on Robotics and Automation, Detroit MI, 1999, 670-675.

[16] K. Sobottka, I. Pitas, Extraction of facial regions and features using colour and shape information, In Proc. of ICIP, 1996, vol. III, 483-486.

[17] M. Sonka, V. Hlavac, Image processing, analysis and machine vision, Chapman & Hall Comp., London, 1994.

[18] J. Triesch, Ch. von der Malsburg, A gesture interface for human-robot-interaction, In Proc. the IEEE 3rd Int. Conf. on Aut. Face and Gesture Recognition, Nara Japan, 1998, 14-16.

[19] D. C. Tseng, C. H. Chang, Color segmentation using perceptual attributes, In Proc. 11th Int. Conf. on Pattern Rec., Den Hague Netherlands, 1992, vol. III, 228-231.

[20] M. A. Turk, A. P. Pentland, Face recognition using eigenfaces, In Proc. of Conf. on Computer Vision and Pattern Recognition, 1991, 586-591.

[21] S. Waldherr, R. Romero, S. Thrun, A gesture based interface for human-robot interaction, Autonomous Robots, vol. 9, 2000, 151-173.

[22] C. Wong, D. Kortenkamp, M. Speich, A mobile robot that recognizes people, In Proc. 7th IEEE Int. Conf. on Tools with Artificial Intelligence, 1995.

[23] G. Xu, T. Sugimoto, A software-based system for realtime face detection and tracking using pan-tilt-zoom controllable camera, In Proc. of Fourteenth Int. Conf. on Pattern Recognition, 1998, vol. II, 1194-1197.

[24] J. Yang, A. Waibel, A real-time face tracker, In 3rd IEEE Workshop on Appl. of Computer Vision, Sarasota Florida, 1996, 142-147.

[25] M. H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: A survey, IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI, vol. 24, no. 1, 2002, 34-58.

Rozpoznawanie gestu wskazującego w oparciu o analizę obrazów z systemu stereowizyjnego

Streszczenie

Artykuł prezentuje algorytm do rozpoznawania gestu wskazującego, który może być szczególnie użyteczny do wskazywania robotowi mobilnemu obiektów zainteresowania. Do rozpoznania poleceń sygnalizowanych przez operatora wykorzystano metody analizy i rozpoznawania obrazów kolorowych powiązane z analizą obrazów stereowizyjnych. W fazie wstępnej algorytmu dokonywana jest detekcja i lokalizacja twarzy. Początek i koniec gestu wskazującego sygnalizowane są za pomocą stosownych gestów wykonywanych ramieniem operatora. System testowano w zadaniach polegających na wskazywaniu i wybieraniu obiektu zainteresowania z grupy obiektów leżących na podłodze. Przetwarzanie i analiza obrazów odbywa się na komputerze pokładowym robota mobilnego.