BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 60, No. 2, 2012
DOI: 10.2478/v10175-012-0044-3

Hand's poses recognition as a mean of communication within natural user interfaces

A. WOJCIECHOWSKI*

Institute of Information Technology, Technical University of Lodz, 215 Wólczańska St., 90-924 Łódź, Poland

Abstract. Natural user interface (NUI) is a successor of command line interfaces (CLI) and graphical user interfaces (GUI) so well known to computer users. A new natural approach is based on extensive human behaviors tracking, where hand tracking and gesture recognition seem to play the main roles in communication. The presented paper reviews common approaches to discussed hand features tracking and provides a very effective proposal of the contour based hand's poses recognition method which can be straightforwardly used for a hand-based natural user interface. Its possible usage varies from medical systems interaction, through games up to impaired people communication support.

Key words: hand's pose recognition, natural user interface, gesture-based interface.

1. Introduction

For years humancomputer interaction was mainly performed by means of mouse and keyboard devices. However, recently, due to computer vision systems and external controllers, natural forms of communication have become more and more popular. Natural user interface is a common parlance, used by system developers, to name intuitive, effective and invisible human-computer communication modes. Contemporary computers can capture, analyze and respond to many human natural communication modes like: voice, hand gesticulation, face mimics, gaze tracking, body language or even brain waves. Besides games and entertainment such systems may be utilized by handicapped or elderly people for communication with environment, also they may be used for controlling mechanical devices when audio communication cannot be applied or direct physical interaction cannot be performed, especially in touchless interfaces useful in healthcare environments [1, 2].

The presented paper concentrates on a computer vision based hands tracking and hands' poses recognition, being one of the most expected and the most extensively developing human-computer communication modes. It is perceived as the most primary and expressive form of human communication. One of the assumptions, for presented system, was its affordability (only popular web camera was used) and real time performance (system should respond at minimum over a dozen frames per second).

Subsequent sections present vision based hand's gesture segmentation and hand recognition methods review. Some of them were adapted and developed to provide robust communication channel that can be used for varied Natural User Interfaces.

2. Overview of the proposed system

The proposed system is based on a camera view analysis and is devoted to track the most recognizable and communicative parts of a human body, like hands and face. After capturing image from the input stream sequence a segmentation process is performed.

For better quality [3] of images segmentation their default RGB color space is modified and images are denoised by appropriate filtering methods [4]. Processed images are searched for human body skin presence and its possible movement. For casually dressed people such a system captures human's hands and face. So recognized body parts can be further analyzed depending on the system eventual use [5]. The implementation tested first, it was deaf-and-dumb sign language communicator where not only relative position of the human body parts should be tracked but occurring gestures must be recognized and analyzed as well. Another implementation, that has not been tested yet, it is a medical interactive visualization system where hand's movement and gestures are used for medical spatial images transformations. Such a vision based, touchless approach lets surgeons perform sterile interaction, while medical operation, with medical visualization system.

3. Hand segmentation

Hand's pose is one of the many human features used for humancomputer communication. As a complex process, human hand detection may cover different hardware and software aspects, however markerless solution, based on simple web camera, seems to be the most valuable and available.

From the software point of view an object recognition encompasses several well tested stages [2]. Thus hand detection process requires image segmentation and its appropriate interpretation. One of the first stages of hand segmentation can be background extraction. The idea behind this method is to subtract the former frame background or initial background from the current image frame. As a result, new or moving object in the image can be detected, however such approach is vulnerable to instable background [6–8] (i.e.: rapid scene illumination changes or swaying trees in the image). Assuming

*e-mail: adam.wojciechowski@p.lodz.pl

that we operate on images sequence I comprising background B and moving or new objects, general background extraction rule can be described with Eq. (1).

$$X_t(s) = \begin{cases} 1 & \text{for} \quad d(I_{s,t}, B_s) > T \\ 0 & \text{for} \quad \text{others} \end{cases} \tag{1}$$

where d is a distance between $I_{s,t}$ – input frame I pixel s on time t and B_s – reference background B model pixel s. T is an acceptance threshold. Main differences between methods lay in background modeling and definition of the d metric.

For a simple method of background subtraction, an absolute value of difference between current frame $I_{s,t}$ pixels and background frame B_s pixels can be calculated (Eq. (2)).

$$X_t(s) = \begin{cases} 1 & \text{for} \quad |B_s - I_{s,t}| > T \\ 0 & \text{for} \quad \text{others} \end{cases} \tag{2}$$

Another simple method considers the difference between corresponding pixels from neighboring, subsequent frames registered respectively in time t and $t-1$ (Eq. (3)).

$$X_t(s) = \begin{cases} 1 & \text{for} \quad d(I_{s,t}, I_{s,t-1}) > T \\ 0 & \text{for} \quad \text{others} \end{cases} \tag{3}$$

Such approach can detect mainly moving objects. It detects perfectly dynamic changes in a scene but it is not able to recognize all pixels belonging to moving object – after frame subtraction object interior is empty. It reflects objects' contour extended into its direction of movement.

Certain improvement to background extraction methods was introduced by Gaussian Mixture Model (GMM) [6] which was an extension to the simple Gaussian model [9]. Its idea is to define k separate Gaussian models for each pixel. As a result each background pixel comprises k different probability distributions. During background detection process source pixel is compared with each of k different Gaussian distributions. If succeeded, a pixel is recognized as a background and impact (weight) of the closest Gaussian distribution is increased (for each pixel, sum of k weights corresponding to its Gaussian distribution set must be summed to 1). If no distribution is assigned to the pixel it is treated as a foreground. Probability density function is calculated according to Eq. (4) and adequate pixel representation, as a weighted sum of several Gaussian distributions, is represented with Eq. (5).

$$G(\xi) = \frac{1}{2\pi\sqrt{|C|}} e^{-\frac{1}{2}(\xi-\mu)^T C^{-1}(\xi-\mu)}, \tag{4}$$

$$p(x,y) = \sum_{i=1}^{k} \omega_i \cdot G_i(\xi), \tag{5}$$

where

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix},$$

$$C = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}, \qquad \omega_{i,t}$$

represent respectively chrominance vector (ξ), mean value (μ), covariance (C) and weight for i-th frame distribution in time $t(\omega_{i,t})$. In Eq. (5) $p(x,y)$ represents pixel in image x,y position, ω_i weight for i-th distribution and $G_i(\xi)$ it is a single probability distribution. According to [10] information about accepted background pixel should be added to the background model. Method treats pixels' values as Gaussian function and compares their mean value $z(i,j)$ according to Eq. (6).

$$J(G_i(\xi), z) > T, \tag{6}$$

where $J(G_i(\xi), z)$ is the Jeffrey's measure specifying whether selected pixel fits $G_i(\xi)$ distribution, and T is a specified threshold. Coefficient $\omega_{i,t}$ is a weight actualized according to Eq. (7).

$$\omega_i = (1-\alpha)\omega_{i,t-1} + \alpha M, \tag{7}$$

where $M = 1$ if pixel belongs to distribution, $M = 0$ otherwise. If ω_i is a small value it can be assumed that its impact to background distribution is relatively small and distribution can be removed. Coefficient α is responsible for learning ability and determines how fast background model is adapted by new pixel.

Another nonparametric *Codebook* [11, 12] method constructs background model basing on input background frames. For each pixel appropriate codebook, consisting of codewords, is created. Each pixel can encompass several codewords depending on its changeability. Pixel's code-words are calculated based on pixel's color and intensity. Each *codeword* c_i consists of color vector v_i and 6-tuple aux_i vector defined with Eq. (8).

$$\begin{aligned} v_i &= (\overline{R}_i, \overline{G}_i, \overline{B}_i) \\ aux_i &= \left\langle \breve{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \right\rangle, \end{aligned} \tag{8}$$

where $\breve{I}_i$ and $\hat{I}_i$ are respectively the minimum and maximum brightness assigned to the codeword, f_i is the frequency the codeword has occurred, λ_i is the longest training period interval that the codeword has not occurred, p_i and q_i the first and the last access time, respectively that the codeword has occurred. Detailed description of the *codebook* construction is presented in [11].

While testing, if source pixel's color does not differ from any corresponding code-words' color more than threshold, and source pixel's intensity falls into code-word's intensity range, it is classified as a background. Otherwise it is assumed to be foreground element. While modeling, each new pixel assigned to the background upgrades appropriate codeword.

The method presented in this paper exploits *codebook* based [12] background subtraction in one of the initial stages. An exemplary hand image (Fig. 1a) after *codebook* based background extraction is presented in Fig. 1d. Further morphological operations and binarization transform the image into a hand binary mask (Fig. 1e).

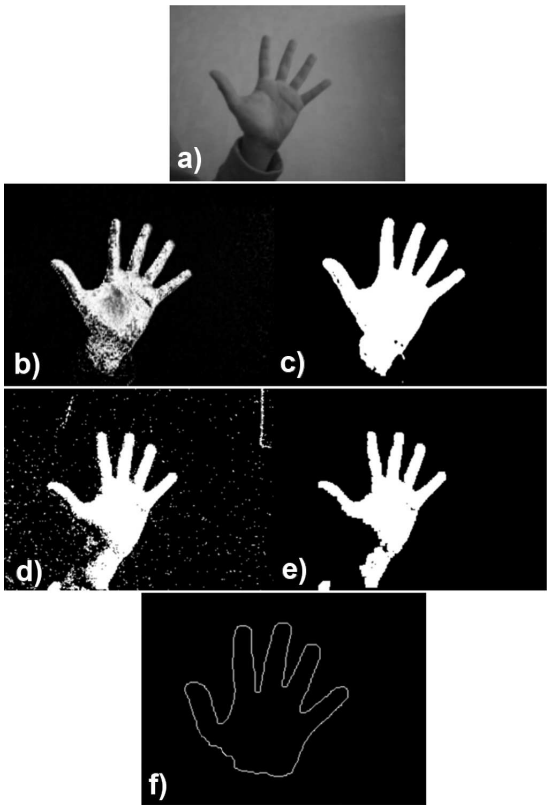

Fig. 1. Hand segmentation stages a) original image with hand; b) image of hand after skin color probability Gauss approximation; c) image b after morphological operations and binarization; d) image of hand after background extraction with Codebook method, Ref. [12]; e) image d after binarization and morphological operations; f) hand's contour made of image c and image e concatenation

A great impact, on effectiveness of hand detection process, has color based segmentation. Goal of a method is to create a decision rule accepting skin pixels and rejecting other ones. Appropriate metric is introduced to measure a distance between reference skin color pixels and others. Besides certain drawbacks (instability in changeable illumination, camera quality dependence, human race) color based approach is a substantial support for hand segmentation. Most of mentioned problems can be eliminated by means of infrared cameras, but their cost is rather high and they were not taken into consideration while research. At the same time selected software refinement was applied.

One of the most basic and effective improvements relies on moving pixels' color assignment into another, equivalent color space. Instead of *RGB* color space a normalized *RGB* space is suggested. New normalized colors (r, g, b) are defined basing on (R, G, B) original values according to Eq. (9).

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}. \quad (9)$$

According to [13] such an approach considerably diminishes troublesome artifacts caused by a changeable light influence or different ethnic groups testing.

Another color segmentation amendment can be achieved by introducing more perceptive color spaces like HSV, HSI or HSL. Their unambiguous interpretation and intuitiveness justifies their common usage in skin classification methods.

One of the examples of skin modeling methods is a direct method encompassing three sets of conditions specified for RGB (0-255) color channels (Eq. (10)).

$$\begin{aligned} &\mathrm{R} > 95, \quad \mathrm{G} > 40, \quad \mathrm{B} > 20 \\ &\max\{\mathrm{R,G,B}\} - \min\{\mathrm{R,G,B}\} > 15 \\ &|\mathrm{R} - \mathrm{G}| > 15, \quad \mathrm{R} > \mathrm{G}, \quad \mathrm{R} > \mathrm{B}. \end{aligned} \quad (10)$$

This method is fast, simple and straightforward, however color space and constant coefficients are chosen empirically and need reconfiguration for different external conditions. More advanced methods can model skin color automatically, but to the detriment of method speed. [14] uses Gauss approximated color histogram. In initial, learning part of the algorithm images comprising just skin color pixels are used to build two chrominance histograms of YUV color space. As introductory histograms do not need to be thorough enough, they are approximated with Gauss normal distribution (Eq. (4)). While testing stage, each pixel is analyzed and retrieved as probability of its distribution fidelity. Image's pixels $z(i,j)$ are transformed according to Eq. (11).

$$z(i,j) = G(\xi), \quad (11)$$

where $G(\xi)$ is a Gaussian distribution from Eq. (4).

The method presented in this paper, besides *codebook* based method, uses color based segmentation as well. Skin color pixels are normalized and modeled with the Gaussian (normal) distribution. As a result, the image with shades of grey, representing pixels skin probability is obtained (Fig. 1b). Further morphological operation and binarization process result in auxiliary binary mask (Fig. 1c). For better results it is performed simultaneously with codebook based background extraction and finally combined into one hand pose contour (Fig. 1f).

4. Hand gesture recognition

In the presented paper the hand gesture analysis and recognition is based on hand characteristic features (i.e.: fingers) detection and tracking. First part of the method comprises color based hand segmentation and image background extraction based on the *codebook* [12] method. An image background extraction method provides a mask (Fig. 1e) encompassing moving objects. It might encompass hands, head or other unexpected objects. Second mask (Fig. 1c) represents only regions with specific (skin) color so it may comprise any body elements.

In second part masks derived from initial stages are binary combined (Fig. 1f). So prepared result mask can be analyzed for hand movements and their features extraction. For a proper hand analysis certain technical descriptors must be calculated. These are (Fig. 2):

- hand's contour defined as a list of contour points $C(i)$;
- minimum bounding rectangle defined with minimum ($x_{\min}$, $y_{\min}$) and maximum ($x_{\max}$, $y_{\max}$) pixels' coordinates in left bottom corner pixel aligned coordinate system;

- centre of mass (*CM*) calculated as a mean value of all hand pixels' coordinates in two image directions;
- hand's bounding circle (circle with centre in hand's centre of mass (*CM*) and radius r as a maximum distance from *CM* to minimum bounding rectangle corners);
- palm bounding circle (circle with centre in centre of mass (*CM*) and radius r_1 equal to 70% of hand's bounding circle radius r – value adjusted experimentally).

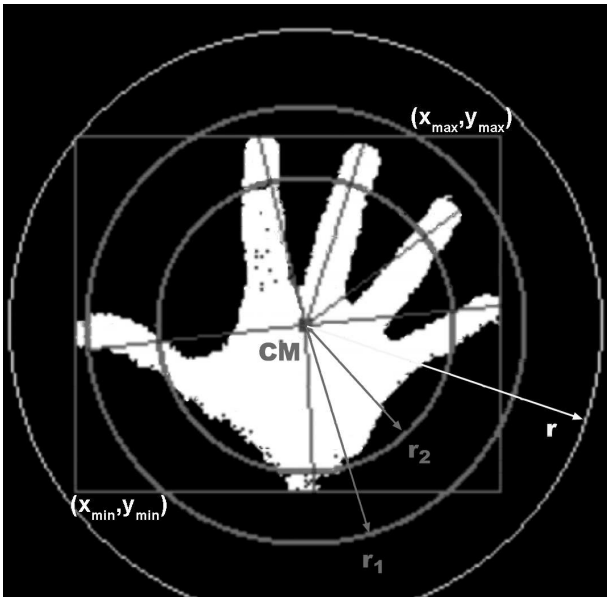

Fig. 2. Set of hand mask descriptors: center of mass CM, hand bounding circle (radius r), hand bounding rectangle, palm bounding circle (radius r_1), smaller palm bounding circle (radius r_2)

Circles were used for number of unbent fingers calculation, and as a result simple hand pose specification.

First approach to hand's fingers detection was calculation of number of objects sticking out of palm bounding circle (radius r_1 in Fig. 2). Unfortunately such approach could detect wrist elements as a finger.

As an improvement another method analyzing hand's contour curvature was introduced. Lets assume that $C(i)$ is an i-th contour point. Then we consider angle θ between two vectors $[C(i), C(i-K)]$ and $[C(i), C(i+K)]$, where K is a constant value. If angle θ exceeds certain threshold then $C(i)$ can be treated as a finger tip. The main difficulty is to evaluate K and θ, since reference contour fluctuates due to background and illumination changes.

The subsequent improvement, providing satisfactory results, consists in measuring distances from the center of mass *CM* and contour points $C(i)$. The chart representing distances for analyzed hand contour is presented in Fig. 3.

Research has proved that two: minimum ($r_2 = 60\%$ of r) and maximum ($r_1 = 75\%$ of r) reference bounding circles detects better unbent fingers. In order to eliminate scarce wrist false positive results, and thumb bad detections additional criteria was suggested. Angle between two vectors $[CM, C(i)]$ and $[CM, C_{r2}(i)]$ was tested, where $C(i)$ was i-th element of the contour and $C_{r2}(i)$ was the closest contour point, to the minimum reference bounding circle (with radius r_2). Then $C(i)$ can be treated as a finger tip if considered angle does not exceed certain value.

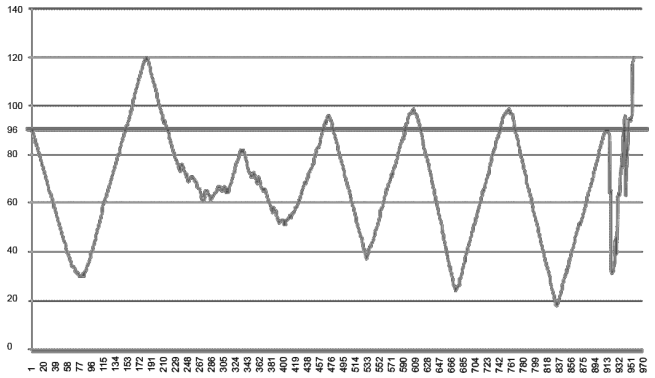

Fig. 3. Curve describing hand's contour distances from the center of mass (*CM*) with horizontal line representing reference palm bounding circle

5. Human body analysis

Besides pure hand recognition certain efforts were put to the whole human posture analysis. Natural User Interfaces may require relative body parts tracking. After color based segmentation and background extraction the system was able to recognize several skin relative spots and elements of environment noise (Fig. 4a).

Disturbances were eliminated using a wide range Gaussian filter (Fig. 4b). In further analysis regions of low importance (spot of number of pixels lower then 100 – image resolution 640×480) were eliminated, as not belonging to the human body. Assuming not destructive environment, no more people in the image, image transformations resulted in maximum three key spots, representing two hands and face but with no information regarding their relative position. In this context, image comprising two or one spot were possible and suggested that two hands or hand and face have overlapped.

System had to be calibrated, before testing with calibrating human posture (Fig. 4) and then initially body parts' position can be retrieved from the image. The largest spot represented face, left minor spot represented right hand and right minor spot represented left hand, so spots area was also important. Further human skin spots tracking was performed for center of mass of each region. Due to spot area permanent changeability, possible noise was reduced with Kalman filtering [15, 16]. It has helped approximate skin spots positions (position of center of mass) following uniformly accelerated motion. However retrieving actual spots' positions, from contentious images consisting of not three but one or two skin spots, was an additional challenge. In such situations, spots area was additionally analyzed. If two relatively wide spots (exceeding certain threshold) were present in the image it meant that two hands were joined and second spot is the face. If one of the spots' areas was relatively similar to one hand initial area, it was interpreted as hand and face overlapping. In case of only one substantial spot the conclusion on all body parts overlapping was made. For contentious situations actual body parts'

centers of mass were not calculated from the image and as a result they were not upgraded and last valid centers of mass were sent to Kalman filter for further interpretation.

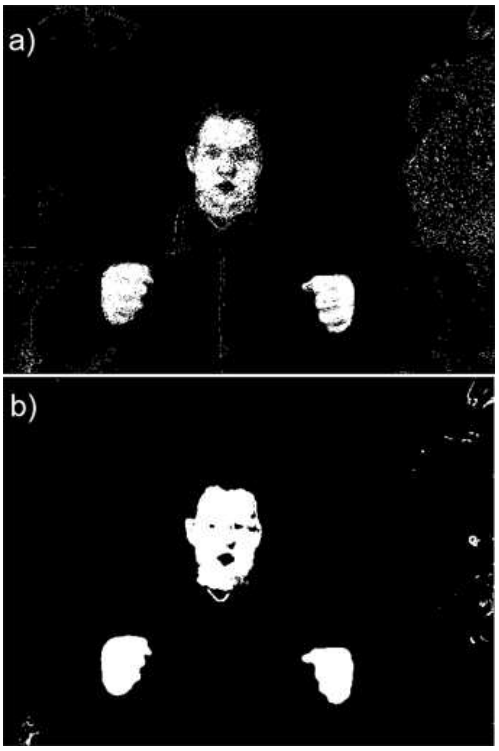

Fig. 4. Binary human posture image a) with visible environmental noise b) without noise due to Gaussian filter of considerable radius

6. Tests

Elaborated system was tested empirically and its functionality allowed performing real-time image analysis basing on an ordinary web camera. In our case *Sony PS-eye* camera was used controlled with *CL-eye* camera driver. System was mainly tested against hand's poses recognition – number of unbent fingers and positions of finger tips were considered. Aspects of human body parts tracking were shortly described, but their exploitation will still be developed, so they do not be discussed in this paper.

As for gestures, set of 8 hand's poses was presented to the system (Fig. 5). Hands' poses were strongly inspired by Polish Sign Language. Each hand's pose was presented to the camera for certain period of time and, while presentation, 20 photos were grabbed from the screen, for each pose. They were used for further statistical analysis.

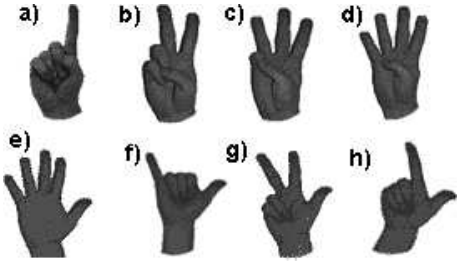

Fig. 5. Exemplary set of hand's poses tested with application

Afterwards hands' poses, captured within camera frames, were analyzed for number of fingers. Visually chosen, properly marked images (like in Fig. 6) were counted. Number of correctly marked images out of 20 captured frames, for each hand's pose, as effectiveness ratio, was collected in Table 1.

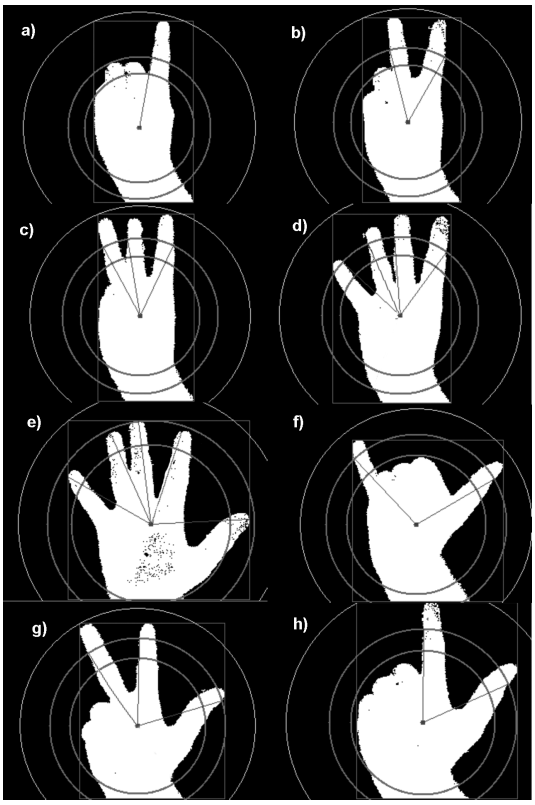

Fig. 6. Correctly recognized hands' poses. Letters a) to h) correspond to hands' poses from Fig. 5. Lines connecting hand's palm center of mass with finger tips mark recognized fingers

Table 1

Hand's poses (Fig. 5) recognition rate. Ratio of correctly marked images out of 20 grabbed images for each hand's pose a) to h).

a)	b)	c)	d)	e)	f)	g)	h)
100%	80%	60%	55%	45%	85%	65%	85%

Application was slightly vulnerable to variant light conditions. It was tested for selected sign language recognition context and its efficiency was very high in a day light environment, however artificial light decreased slightly its efficiency. Hand's pose was recognized properly if number of unbent fingers was equal to number of lines connecting hand's palm center of mass with finger tips (Fig. 6). Unfortunately changeable light conditions and relatively not professional camera lens resulted in image flickering. As a consequence several mistakes took place while testing. It has happened that false wrist recognition appeared (Figs. 7a,b,c,d) or not satisfactory number of finger tips was detected (Figs. 7e,f,g,h).

Described tests revealed an average method effectiveness of 72%. Individually hands' poses, presented in Fig. 5, achieved different recognition rates presented in Table 1.

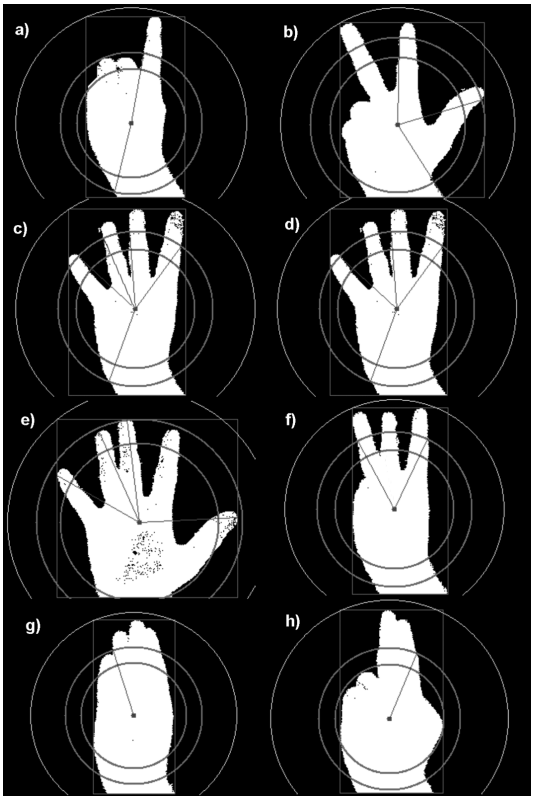

Fig. 7. Exemplary faults have appeared while hands' poses testing. a) – d) false wrist recognition; e) – h) too little finger tips recognized

It must be noticed that recognition rate decreased with growing number of fingers. Single finger was recognized without any problems, however in case of five fingers more then half of captured frames contained unintended faults. Probably one of the sources of low recognition rate came from overlapping fingers caused by changeable hand orientation in relation to the camera – method has detected several joined fingers as one finger (Figs. 7g, 7h). Additionally presented solution was got rid of any depth data so camera could not track bent fingers, which were not visible on the hand's palm background.

7. Conclusions

Summing up this paper presents really effective, available and simple human features tracking method. It introduces novice gesture recognition contour based approach featuring considerable effectiveness. System was tasted against sign language hands' poses recognition, but its functionality seems to be universal and, according to author, it can be used for intuitive game interface creation or even sophisticated medical visualization environment interaction.

Further system development may concern its calibration. At present it is calibrated once, at the beginning of the program, but due to permanent environment changes it decalibrates. To make the system more ergonomic, it can be automatically calibrated or recalibrated basing on Haar detected face skin method. Some more advanced methods of human body tracking, when they overlap each other or just simply disappear, can be also incorporated in the project. Motion tracking or motion prediction methods seem to be useful for further research [17, 18].

Better fingers segmentation can be achieved by means of Microsoft Kinect device, which is not only capable of color image recording, but provides additional depth data indispensable for bent or overlapping fingers tracking as well.

REFERENCES

[1] J.P. Wachs, M. Kolsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications", *Communication ACM 02/2011* 54 (2), 60–71 (2011).
[2] R. Tadeusiewicz and M.R. Ogiela, "Structural approach to medical image understanding", *Bull. Pol. Ac.: Tech.* 52 (2), 131–139 (2004).
[3] K. Guzek and P. Napieralski, "Measurement of noise in the Monte Carlo point sampling method", *Bull. Pol. Ac.: Tech.* 59 (1), 15–19 (2011).
[4] P. Lipinski and M. Yatsymirskyy, "On synthesis of 4-tap and 6-tap reversible wavelet filters", *Electrotechnical Review* 84 (12), 284–286 (2008), (in Polish).
[5] Sz. Myśliński, "On the generation of graph representation of hand postures for syntactic pattern recognition", *J. Applied Computer Science* 17 (1), 71–84 (2009).
[6] G. Jianfeng, C. Jingzhu, S. Xuehua, and C. Yu, "A robust moving objects detection algorithm based on Gaussian mixture model", *Proc. ITCS 2009* 1, 566–569 (2009).
[7] Y. Liang, S. Guo, and Z. Wang, „A robust and fast motion segmentation method for video sequences, *Proc. IEEE Int. Conf. on Automation and Logistics* 1, 2952–2957 (2007).
[8] L. Huang, Z. Yu, Y. Yu, and C. Zhou. "A moving target detection algorithm based on the dynamic background", *Proc. CiSE 2009* 1, 1–5 (2009).
[9] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19 (7), 780–785 (1997).
[10] Y. Liu and P. Zhang, "Vision-based human-computer system using hand gestures", *Proc. CIS'09* 1, 529–532 (2009).
[11] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model", *Elsevier Real-Time Imaging* 11, 172–185 (2005).
[12] M. Fathy and M.H. Sigari, "Real-time background modeling/subtraction using two layer codebook model", *Proc. IMECS* 1, CD-ROM (2008).
[13] A. Andreeva, V. Vezhnevets, and V. Sazonov, "A Survey on pixel-based skin color detection techniques", *Proc. GraphiCon* 1, CD-ROM (2003)
[14] M. Wysocki, T. Kapuściński, J. Marnik, and M. Oszust, *Hand Gestures' recognition in a Vision System*, *Rzeszów University of Technology Publishing House*, Rzeszów, 2011, (in Polish).
[15] N. Funk, *A Study of the Kalman Filter applied to Visual Tracking*, University of Alberta, Alberta, 2003.
[16] K. Szabat, T. Orlowska-Kowalska, and K.P. Dyrcz, "Extended Kalman filters in the control structure of two-mass drive system", *Bull. Pol. Ac.: Tech.* 54 (3), 315–325 (2006).
[17] S. Denman, C. Fookes, and S. Sridharan, "Group segmentation during object tracking using optical flow discontinuities", *Symp. on Image and Video Technology* 1, 270–275 (2010).
[18] I. Dulęba, "Impact of control representations on efficiency of local nonholonomic motion planning", *Bull. Pol. Ac.: Tech.* 59 (2), 213–218 (2011)