CONSTRAINED CONTOUR MATCHING IN HAND POSTURE RECOGNITION

ARTUR WILKOWSKI[12] WŁODZIMIERZ KASPRZAK[1]

[1] Institute of Control and Computation Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw Poland

[2] Faculty of Geodesy and Cartography, Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warsaw Poland

A.Wilkowski@ia.pw.edu.pl, W.Kasprzak@ia.pw.edu.pl

Abstract. In this paper constrained contour models are applied for hand posture recognition in single color images. In particular, the proposed algorithm utilizes a class of physics-based modelling methods called *Deformable Templates* [1],[2],[3]. After color-based image segmentation a contour hypothesis is detected and some features are extracted, suitable for comparison with the template's geometric properties. Several metrics for matching contour templates against image data are discussed. The described methods are evaluated experimentally and referred to a known hand posture recognition algorithm.

Key words. image processing, hand posture recognition, gesture recognition, deformable templates

1 Introduction

Due to a massive increase of tasks performed by computers in our daily life the problems of human-computer communication and human-computer interaction become crucial in modern computer systems. A continuous development of computer technologies and steady increase of computer efficiency opens new horizons in this field. Nowadays the human-computer interaction (HCI) research requires a new approach in constructing interfaces between a human and machine. The gestures provided by a human body form an important part of natural communication and the development of gesture recognition systems is therefore a natural consequence of trends in modern HCI. The gestures can be performed using different means of expressions, we will concentrate on the ones performed by a hand - a hand image is continuously captured by a camera, transmitted to the computer and on-line processed there. We shall not consider any support material, common to haptic devices, such as data gloves [4].

While gestures are inherently dynamic processes, the problem of gesture recognition in image sequences requires first a nearly independent processing of each single image, leading either to the recognition of a static hand pose [5] or to some kind of a good numerical or symbolic description of the hand object [6]. Thus obtained data can then be used as input of the hand gesture recognition stage, together with some appropriate method of dynamic system modelling, eg. Hidden Markov Models [7] or Dynamic Bayesian Networks [8] - some important probability-based modelling methods.

Current paper concentrates on the single image recognition stage. One can distinguish two basic categories of hand posture recognition methods - the model-based recognition and model-less classification. In the latter case the recognition process consists of two data-driven phases: image segmentation with numerical features extraction and feature vector classification. The segmentation stage often explores skin color distribution for background subtraction, but it also uses edge and motion information for foreground object detection ([5],[6]). The contour of the hand can be characterized using geometrical moments or spectral features ([9],[10]) or other numeric features of the contour like orientation of edges [11] or local maxima of the contour's distance-to-mass-center function [5].

In case of model-based methods the distinction between segmentation and description/recognition phases becomes less clear. A symbolic model can be applied to a pre-segmented image, but the model can also directly support the image segmentation stage. Although both 3D and 2D models of objects are possible, it is often argued that 3D pose estimation is a highly computationally complex process and in practice 2D models are utilized more often. Variety of approaches based on 2D hand models encompass methods based on scale-space blob representation [12], probabilistic models [13] or elastic graph matching [14]. Some other related methods - anatomically-inspired graph models of body parts usually suffer from significant computational complexity of subgraph search [15],[16].

Model-based approaches in general can be more resistant to image clutter than the bottom-up classification approaches, but at the expense of requiring larger computation time.

This paper proposes a deformable template matching method for the solution of the hand posture recognition in single images. Particular elements of this approach are explained in sections 2 - 5, whereas the approach is summarized in section 6.

Image segmentation is mainly based on skin color analysis (section 2). The model shape is given by a B-Spline curve (section 3). The matching procedure looks for the most similar object within a space of allowed Euclidean transformations (section 4). Current image segments are used for spline initialization and they form a measurement basis for contour matching and computation of the matching error (section 5). After all elements of this approach are known the contour fitting algorithm is finally specified in section 6. Several error measures are proposed and tested in the experimental part(section 7).

2 Image preprocessing and segmentation

Feature vectors used for classification of hand postures are obtained by a geometrical analysis of the area in the image covered by hand. In the proposed approach the segmentation of the hand is based on color analysis and detection of skin color in the image.

The traditional RGB color space is not suitable for this task, as all its 3 component values depend on the lighting conditions. Much more convenient for image analysis are color-spaces that operate separately on chrominance and luminance values. The skin color is detected primarily on base of chrominance values - a change in light intensity (e.g. due to shadows) does not affect too much the results of segmentation.

The YCbCr color space is applied for image segmentation in this work. The matrix that can be used for transformation between RGB and YCbCr is given in the following equation (all color values are assumed to be in the range $[0, 255]$).

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -1.687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1)$$

The resulting color space can then be a subject to statistical modelling of skin color distribution based on previously gathered samples [10]. However, in the proposed implementation static thresholds for skin color detection are used [6],[17]:

$$85 \leq Cb \leq 135,\ 135 \leq Cr \leq 180 \quad (2)$$

The proposed thresholds proved to ensure acceptable results for most samples used in the experiments.

(a) Original image (b) Image after color processing (c) Image after edge extraction

Fig. 1: Stages of image preprocessing

Note that the luminosity was neglected completely in the thresholds adopted.

Due to inevitable errors in classification of the skin color the resulting binary image has usually quite irregular shape (e.g. contains unwanted holes inside). This can have a bad impact on contour fitting. Therefore, after the image binarization step, the morphological operation of *n-closure* is applied (it contains an n times *erosion* operation preceded by n times *dilation*) to regularize the resulting shape (in our experiments the value $n = 1$ was adopted).

After performing morphological operations - the largest continuous object in the picture is chosen as a hand image (the N_4 neighborhood is used). Afterwards the largest continuous object in the picture is chosen as a possible location of the hand in current image. Such prepared binary image is used for creating the contour image, that contains the object outline (boundary). Image preprocessing steps are illustrated by sample results in Fig. 1.

3 Contour model

3.1 B-Spline hand model

In our description of the modelling of rigid objects using B-splines we follow the book [2]. For the purpose of hand pose recognition, the object is represented as a spline founded on the circular basis of quadratic spline functions. The template spline is described by a set of ordered control points, and is a linear combination of basis functions, where coordinates of control points act as weights (Fig. 2a). The spline is defined independently for both x and y coordinates. The template spline can be described by the equation:

$$x(s) = \sum_{n=0}^{N_b-1} q_x^n B_n(s), \tag{3}$$

where N_b is the number of functions in the basis, q_x^n is the value of the n-th control point and $B_n(s)$ is the value of the n-th spline basis function at point s. More compact matrix notation for this equation is:

$$x(s) = \mathbf{B}(s)^T \mathbf{Q^x}, \tag{4}$$

where $\mathbf{B}$(s) is a vector of values for all spline functions in the basis, while $\mathbf{Q^x}$ denotes a control vector made up of x control point coordinates, i.e.

$$\mathbf{Q^x} = \begin{pmatrix} q_0^x \\ \dots \\ q_{N_b-1}^x \end{pmatrix} \tag{5}$$

A 2-dimensional spline is constructed as a composition of two independent 1-dimensional splines founded on the same spline basis:

$$\mathbf{r(s)} = (x(s), y(s)) \tag{6}$$

In the compact matrix notation the 2-dimensional spline can be defined as

$$\mathbf{r}(s) = U(s)\mathbf{Q}, \tag{7}$$

where $\mathbf{r}(s)$ is a vector (x, y) of spline coordinates for the given spline parameter s, $U(s)$ is a matrix defined as

$$U(s) = \begin{pmatrix} \mathbf{B(s)^T} & \mathbf{0} \\ \mathbf{0} & \mathbf{B(s)^T} \end{pmatrix} \tag{8}$$

and Q is a double-length control vector made up of individual control vectors for the x- and y-dimension:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q^x} \\ \mathbf{Q^y} \end{pmatrix} \tag{9}$$

In the space of spline functions there can be easily defined an inner product in form

$$< x, y >= \frac{1}{L} \int_{s=0}^{L} x(s)y(s)ds, \tag{10}$$

where L is the maximum value of spline parameter (depending on the number of spline segments used). This inner product induces the so called L_2 norm useful for calculating similarities between curves. For a 1-dimensional spline the norm is defined as

$$\|x\|^2 = \frac{1}{L} \int_{s=0}^{L} x(s)^2 ds \tag{11}$$

while for a 2-dimensional curve we have:

$$\|\mathbf{r}\|^2 = \frac{1}{L} \int_{s=0}^{L} |\mathbf{r}(s)|^2 ds \tag{12}$$

An efficient way of calculating the L_2 norm is provided in [2]. The norm for the vector of control points is here defined to be equal to the norm of the associated spline. In one dimension we have:

$$\|\mathbf{Q^x}\| = \|x\| \tag{13}$$

Thus the norm for a vector of control points can be calculated as

$$\|\mathbf{Q^x}\| = \sqrt{(\mathbf{Q^x})^T \mathcal{B} \mathbf{Q^x}} \tag{14}$$

where $\mathcal{B}$ is called the *metric matrix* and it depends on the spline basis used:

$$\mathcal{B} = \frac{1}{L} \int_{s=0}^{L} \mathbf{B}(s)\mathbf{B}(s)^T ds \tag{15}$$

Equivalently, in the 2-dimensional space we specify the norm

$$\|\mathbf{Q}\| = \|\mathbf{r}\| \tag{16}$$

Hence the norm for a vector of control points is computed as

$$\|\mathbf{Q}\| = \sqrt{(\mathbf{Q})^T \mathcal{U} \mathbf{Q}}, \tag{17}$$

where $\mathcal{U}$ is a matrix related to $\mathcal{B}$:

$$\mathcal{U} = \begin{pmatrix} \mathcal{B} & 0 \\ 0 & \mathcal{B} \end{pmatrix} \tag{18}$$

3.2 Spline approximation

One way of defining the spline is by specifying its control points. An even more interesting method is to define the spline as the approximation of some number of measurements obtained from the image. If the measurements can be represented in a parametric form as a function of some parameter, eg. $f(s)$ (the derivation of parameter values during template initialization will be presented later on), the spline function f^*, which approximates the unknown contour function f, can be obtained by knowing that the following equation holds:

$$< f - f^*, B_i >= 0, \forall i = 0, ..., N_b - 1 \tag{19}$$

The above formula leads to a set of N_b linear equations. However, to obtain the right-hand-side inner products, a numerical integration must be applied. The f^* spline obtained in this way is of arbitrary shape (limited only by the form of its spline basis) but it is the best approximation of the f function with respect to the L_2 norm.

4 Shape space

In the approach presented here the single hand posture is represented by a template spline. In image, however, the corresponding posture will be of slightly different shape. At least it will differ in terms of its position, rotation and scale. Thus according to methodology proposed in [2] the final (transformed) shape can be represented by the equation

$$\mathbf{Q} = W\mathbf{X} + \mathbf{Q_0}, \tag{20}$$

where $\mathbf{Q_0}$ is the original template shape, $\mathbf{Q}$ is the shape after transformation, $\mathbf{X}$ is the parameter vector and W is the transformation matrix associated with a group of allowed transformations. Different values of the parameter vector $\mathbf{X}$ form the so called *shape space*. In our experiments the space of Euclidean similarities has been chosen. The transformation matrix for the shape space of Euclidean similarities is specified as follows

$$W = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{Q_0^x} & -\mathbf{Q_0^y} \\ \mathbf{0} & \mathbf{1} & \mathbf{Q_0^y} & \mathbf{Q_0^x} \end{pmatrix} \tag{21}$$

Thus having a shape approximated from image measurements (obtained by methods described earlier) $\mathbf{r_f}$ and an associated control vector $\mathbf{Q_f}$, one can find a globally optimal matching, in terms of the L_2 norm, between the template shape and the "measured" shape by optimizing the expression:

$$\min_{\mathbf{X}} \|W\mathbf{X} + \mathbf{Q_0} - \mathbf{Q_f}\|^2 \tag{22}$$

A closed-form solution of above equation (22) can be computed using the pseudo-inverse matrix W^+ (as discussed with more details in [2])

$$\widehat{\mathbf{X}} = W^+(\mathbf{Q_f} - \mathbf{Q_0}). \tag{23}$$

The matrix W^+ is expressed by the following formula

$$W^+ = \mathcal{H}^{-1}W^T\mathcal{U}. \tag{24}$$

$\mathcal{H}$ can be interpreted as a metric matrix for parameters from the *shape space* and which can be derived as

$$\mathcal{H} = W^T\mathcal{U}W. \tag{25}$$

The optimization process described above can be also viewed as a projection of the measured spline onto the *shape space*. Obviously the solution obtained is only approximately optimal due to the local nature of given measurements and noise. Therefore some later iterative improvement may be desirable.

5 Contour-to-model matching

5.1 Contour initialization

Important part of the hand object recognition procedure is the initial estimation of contour parameters. A properly initialized image contour can form a good starting point for consecutive measurements and successful match with the proper template. In particular, after calculation of a binary image by methods described in sec. 2 the following geometrical moments are computed:

- object area A,
- object centroid position (c_x, c_y), and
- the inertia matrix $\mathcal{I}$.

In order to obtain a good correspondence of template contour moments to geometrical moments of the image patch, all possible holes in the prospective hand area in binary image are filled first and the moments are computed next.

These geometric moments are used to initialize shape space parameters in case of using space of Euclidean similarities.

- The displacement factor equals to object centroid position (providing that a template has been centered around point (0, 0)),
- The scaling factor k can be computed as a root square of A/A_0, and
- the rotation angle θ is computed from the rotation of the object's main geometric axis, which equals to the largest eigenvector of the inertia matrix $\mathcal{I}$. Obviously, in order to compute the rotation also the inertia matrix for the original

template spline $\mathcal{I}_0$ must be known if the object has not been normalized beforehand.

The parameters for the *shape space* of Euclidean similarities can be then set up as

$$[c_x, c_y, k\cos\theta - 1, k\sin\theta]. \tag{26}$$

The necessary geometrical moments for the template spline - the area A_0 and inertia matrix $\mathcal{I}_0$ - can be quite easy computed by numerical integration.

5.2 Measurements

The measurements required for spline approximation described in sec. 3.2 are again computed from the segmented image, obtained in sec. 2. We make a simplifying assumption that the image contains a simple contour. First the *Gaussian blur* followed by the *Sobel mask* in x and y directions is used, then absolute values for pixels in both directions are calculated and such prepared images are summed together forming the edge image.

According to [2], in order to perform measurements some initial approximation of a spline should be available. This can be done in two ways: by utilizing the initialization of geometrical moments from section 5.1 or by taking the already matched curve using methods from section 4.

In any case the measurements are searched along initial spline normal vectors. The vectors are distributed evenly along the spline with respect to the spline parameter s. The search is performed in both directions looking for the strongest image response (such as presence of an edge), see Fig. 2b.

A bi-linear filtration is used as an anti-aliasing measure. If no measurement along normal vector is found, the point on the initial spline is used instead (this is to avoid numerical singularities in approximation).

5.3 Measure of fitness

An important requirement in presented recognition approach is to use some measure of fitness, i.e. to evaluate how well a template matches the image data. The most straightforward measure is based on the L_2 norm. A curve, that is initialized using geometrical moments, is the starting point in the curve approximation step (see section 3.2). The two curves, the image-based and the template one, are then compared using the L_2 norm. One major flaw of this approach is that it gives no good way to handle the lack of measurements.

Another possible fitness measure, that we decided to test, is based directly on measurements taken from the initialized curve. The error measure can be defined as follows:

$$e_1 = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{r}(s_i) - \tilde{\mathbf{r}}(s_i)|^2 \tag{27}$$

where $\tilde{\mathbf{r}}(s_i)$ is the measurement point found along the normal vector originating from point $\mathbf{r}(s_i)$ on the initialized template curve and N is the number of measurements. If no measurement along curve normal could be found a penalty value is applied instead. The main possible weakness of this measure lies in the curve parametrization method. This is because the normal vectors used for performing measurements are evenly distributed along curve with respect to the curve parameter, and not necessarily with respect the to actual distance between vectors (which depends on the selection of control points).

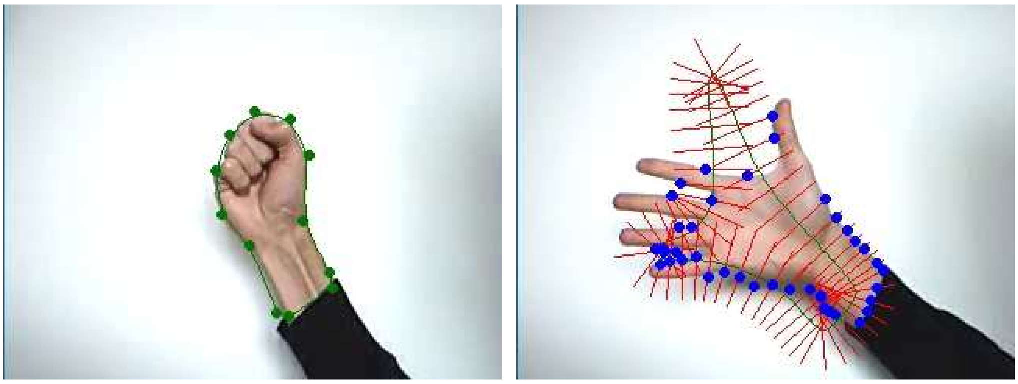
Therefore we propose a modified measure that takes into account differences in curve segment lengths

$$e_2 = \left(\sum_{i=0}^{L-1} N_i d_i\right)^{-1} \sum_{i=0}^{L-1} d_i \sum_{j=1}^{N_i} |\mathbf{r}(s_{i_j}) - \tilde{\mathbf{r}}(s_{i_j})|^2 \tag{28}$$

where N_i is the desired number of measurements within a unit spline interval, d_i is a curve length within a unit spline interval and s_{i_j} is the curve parameter value in the i-th curve segment. The unit arc length d_i is the result of the integration

$$d_i = \int_{s=i}^{i+1} |\mathbf{r}'(s)| ds \tag{29}$$

which can be computed numerically off-line, similar to d. As can be easily seen this approach is very similar to the previous one, except that it applies weight



(a) (b)

Fig. 2: Contour features: (a) a B-spline contour with control points, (b) contour normal vectors and measurements.

factors that prefer measurements coming from longer spline segments.

Another related approach, that also takes into account varying densities of measurement distribution along the spline, is based on a weighting system suggested in [2]. It can be represented as a following weighted sum

$$e_3 = \left(\sum_{i=1}^{N} w_i \right)^{-1} \sum_{i=1}^{N} w_i |\mathbf{r}(s_i) - \tilde{\mathbf{r}}(s_i)|^2. \qquad (30)$$

The weights correspond to the magnitude of the curve's first derivative vector

$$w_i = |\mathbf{r}'(s_i)| \qquad (31)$$

In this approach larger weights are attributed to sparsely distributed measurements. However, this time weighting is performed on a per-measurement basis.

6 Template fitting algorithm

Let us now summarize our approach to hand posture recognition in a single image. It it split into the *modelling* phase and the recognition stage.

The modelling phase:

1. Each of the hand postures that need to be recognized is encoded as a vector of B-spline control points (sec. 3.1)

The recognition algorithm works as follows:

1. For each image a binary hand region is obtained using image segmentation described in sec. 2

2. The region is used to initialize parameters for every template spline (they correspond to different hand postures) using geometrical moments (sec. 5.1)

3. Every one of the initialized template splines forms a basis for taking measurements along the normal-to-curve vectors (sec. 5.2)

4. A new spline is approximated from measurements (sec. 3.2) and this spline is projected onto the *shape space* of the template spline, thus improving the initialization parameters from the previous step (sec. 4). The fitting steps can be repeated several times (or omitted at all) to find the best template spline fit.

5. The best fitted template spline is evaluated in terms of its actual correspondence with the image using measures specified in sec. 5.3. A template spline with the lowest error is chosen as the result of classification.

Tab. 1: Recognition rates for different matching methods

similarity measure	recognition rate
L_2	90.0%
e_1	90.4%
e_2	87.1%
e_3	90.2%
e_1 + fitting	88.4%
color-based	99.9%

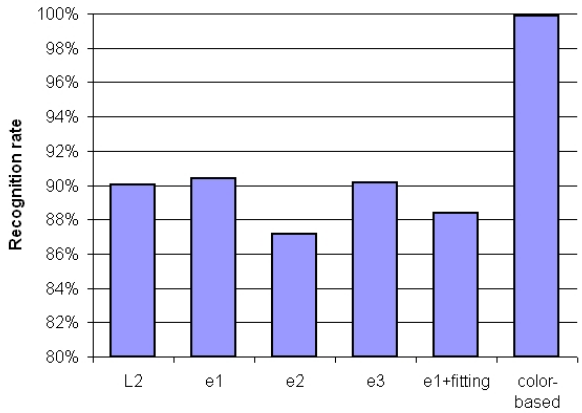


Fig. 4: Recognition results for different fitness measures

7 Experimental results

In order to evaluate the approach proposed in current paper a set of 9 different hand poses was utilized. In order to simplify the color segmentation process it was assumed that the hand performing gestures was wearing a long dark sleeve. The set of hand poses used is presented in Fig. 3.

For the recognition process a hand pose sequence of 24 hand posture images was available, with image resolution of $320x240$ pixels. In total 1462 frames were evaluated independently, transitions between hand poses were neglected.

Six different classification methods were used in the comparison. Four of them were based on fitness measures proposed in sec. 5.3:

1. the L_2 norm,
2. the e_1 measure,
3. the e_2 measure,
4. the e_3 measure

The fifth measure was a e_1 measure preceded by three iterations of spline fitting algorithm (sec. 4). For comparison also a color-only based classifier from [6] was used as the sixth method.

The recognition rates for all these methods are given in table 1 and Fig. 4.

The results indicates that the best fitness measure for the experimental data set was the e_1 measure, which reached nearly 90.4% of recognition accuracy. Most other methods achieved results close to 90%. The fittness measures that utilize weighing gave average (e_3) or little below average (e_2) results. The recognition rate for the reference color-based classifier is still the highest on - 99.9%. This can be explained by the fact that the latter algorithm is especially suited to accommodate natural variability in hand shapes.

It also turns out that the additional utilization of the fitting algorithm together with the e_1 measure does not increase the recognition rate, it shows even several percents lower accuracy. The possible reason for it - due to more fitting iterations the algorithm is more sensitive to image noise and inaccuracy of hand pose presentation (in comparison to the more robust method of moments-based initialization).

The largest part of errors in the experiments emerged from mistaking hand pose no. 3 and no. 7 (see: Fig. 3). Hand pose no. 3 is characteristic due to its complicated shape and significant variability of possible utterances, which both can strongly affect the recognition accuracy. It is also more difficult to establish the main geometrical axis for this pose's shape, which negatively influences the accuracy of moment-based hand orientation estimation (see Fig. 5).

A non-optimized version of the algorithm using the e_1 measure processes about 20-50 frames per second, depending on the PC processor type used.

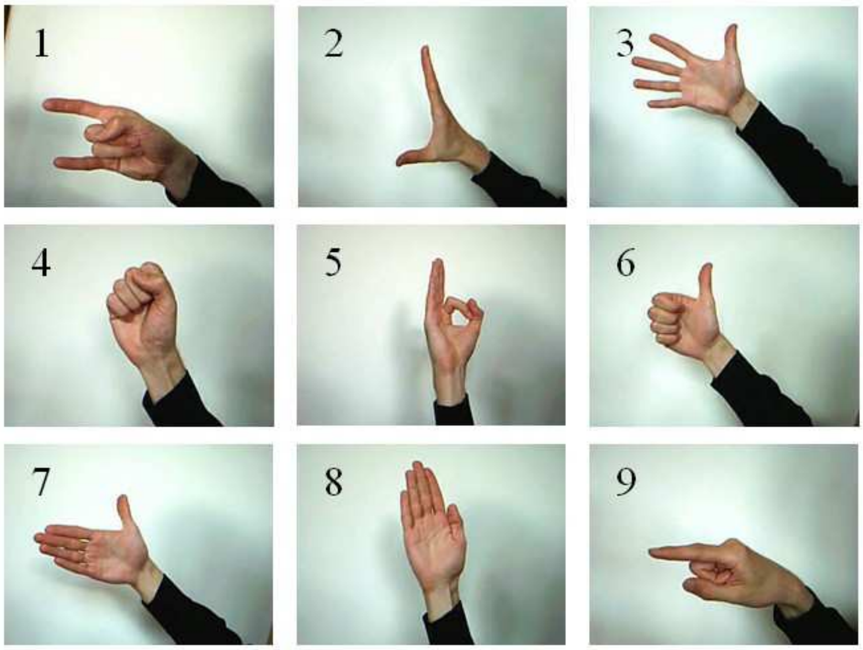


Fig. 3: A set of recognized hand poses

Fig. 5: An example of imperfect hand orientation estimation

8 Conclusions

In this paper a method of hand posture classification in single images was proposed and tested. It utilizes hand shape models according to the Deformable Templates principle.

The experiments prove that the modelling method, although oriented on rigid objects, can also be used to recognize non-rigid ones (like a human hand), providing that some suitable discrete versions of allowed shape classes are prepared in advance.

The method can be easily extended to use the object tracking mechanisms developed for Deformable Templates [2] making it more robust and resistant to clutter. The measures of fitness proposed prove to be adequate for the recognition task and ensure good recognition rates.

Acknowledgements This work was supported by the Polish Ministry of Science and Higher Education by the grant N N514 1287 33.

References

[1] Terzopoulos D. Deformable Models: Classic, Topology-Adaptive and Generalized Formulations. In: *Geometric level set methods in Imaging, Vision, and Graphics.* Springer-Verlag, New York, pages 21-40, 2003

[2] Blake A, Isard M. *Active Contours.* Springer-Verlag London, 1999

[3] Kass M, Witkin A, Terzopoulos D. Snakes. Active contour models. *International Journal of Computer Vision*, 1:321-331, No. 4, 1998

[4] Lee C, Xu Y. Online, Interactive Learning of Gestures for Human/Robot Interfaces. *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, pages 2982-2987, 1996

[5] Kasprzak W, Skrzyński P. Hand image interpretation based on double active contour tracking. In: *ROMANSY 16. Robot design, dynamics, and control*, CISM Courses and lectures - No. 487, Springer, Wien NewYork, pages 439-446, 2006

[6] Wilkowski A. An Efficient System for Continuous Hand Posture Recognition in Video Sequences. In: *Computational Intelligence: Methods and Applications (Challenging Problems of Science series)*, EXIT Warsaw, pages 411-422, 2008

[7] Rabiner L R. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE.* 77:257-286, No. 2, 1989

[8] Murphy K P. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD Thesis, UC Berkeley, 2002

[9] Kauppinen H, Seppanen T, Pietikainen M. An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17:201-206, No. 2, 1995

[10] Kapuscinski T. *Recognition of the Polish Sign Language in a Vision System* (in Polish). PhD Thesis, University of Zielona Gora, 2006

[11] Freeman W T, Roth M. Orientation Histograms for Hand Gesture Recognition. *Proceedings of the IEEE International Workshop on Automatic Face and Gesture Recognition* Zurich, pages 296-301, 1995

[12] Laptev I, Lindeberg T. Tracking of multi-state hand models using particle filtering and a hierarchy of mutli-scale image features. *Technical Report CVAP* 245, 2000. ISRN KTH/NA/P-00/12-SE

[13] Moghaddam B, Pentland A. Probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19:696-710, No. 7, 1997

[14] Triesch J. *Vision Based Robotic Gesture Recognition.* PhD thesis, Ruhr Universitat Bochum, 1999

[15] Felzenszwalb P, Huttenlocher D P. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61:55-79, No.1, 2005

[16] Sigal L, Black J. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 2, pages 2041-2048, 2006

[17] Kukharev G, Nowosielski A. Visitor Identification - Elaborating Real Time Face Recognition System. *Proceedings of WSCG'2004 Short Communications*, Plzen, Czech Republic, pages 157-164, 2004