VERSITA

Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 2, 437–448
DOI: 10.2478/v10006-012-0033-6

amcs

HAND GESTURE RECOGNITION BASED ON FREE–FORM CONTOURS AND PROBABILISTIC INFERENCE

WŁODZIMIERZ KASPRZAK [*], ARTUR WILKOWSKI [*,**], KAROL CZAPNIK [*]

[*] Institute of Control and Computation Engineering
Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: W.Kasprzak@elka.pw.edu.pl,
K.Czapnik@stud.elka.pw.edu.pl

[**] Faculty of Geodesy and Cartography
Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warsaw, Poland
e-mail: A.Wilkowski@elka.pw.edu.pl

A computer vision system is described that captures color image sequences, detects and recognizes static hand poses (i.e., "letters") and interprets pose sequences in terms of gestures (i.e., "words"). The hand object is detected with a double-active contour-based method. A tracking of the hand pose in a short sequence allows detecting "modified poses", like diacritic letters in national alphabets. The static hand pose set corresponds to hand signs of a thumb alphabet. Finally, by tracking hand poses in a longer image sequence, the pose sequence is interpreted in terms of gestures. Dynamic Bayesian models and their inference methods (particle filter and Viterbi search) are applied at this stage, allowing a bi-driven control of the entire system.

Keywords: active contours, hand pose detection, hand tracking, image sequence analysis, stochastic inference.

1. Introduction

Hand image recognition is mainly considered in the context of man-machine communication (Rehg and Kanade, 1993) and person identification (biometrics) (Sanchez-Reillo *et al*., 2000). Here we focus on the first application area. The system's main attribute is given by the fact whether it is designed for static posture recognition or for (dynamic) gesture recognition. A useful application of the first type of systems would be the recognition of a "finger alphabet" (Marnik, 2003) or the Polish sign language postures (Flasiński and Myśliński, 2010), as applied in mute people's communication.

Current research on gesture recognition in image sequences concentrates on the flexibility of system use: (i) on "freely" located hands, without gloves, in front of a random structured background, and (ii) on the general-purpose architecture of the recognition system, i.e., declarative languages for model representation, learning and recognition. The main motivation for such research is to make the man-machine interface more flexible and easier for the user.

The "free hand" image-based gesture recognition process can be decomposed into three main stages:

- single frame preprocessing and segmentation,
- hand feature extraction and pose classification, and
- hand tracking and pose sequence (gesture) interpretation.

For the image segmentation task a large number of different computational approaches and algorithms already exist (e.g., Niemann, 2000; Pitas, 2000; Gonzalez and Wintz, 1987; Kasprzak, 2009). As the human skin color is a very characteristic feature, in our approach we shall provide a color-based region-of-interest detection (Yining *et al*., 1999; Fu *et al*., 2000; Emambakhsh *et al*., 2010).

A straightforward approach to object detection is to extract its shape (or contour) information from the image. Most often shape recognition is accomplished with appearance-based methods (Rafajłowicz *et al*., 2008), although an explicit shape model could also be used here (Emambakhsh *et al*., 2010). A popular approach to closed contour detection for free-form and deformable objects uses *active contours* (Kass *et al*., 1998) or *snakes*

(Terzopoulos, 2003). Snakes (active contours) are curves defined in the image plane that can change their shape and move under the influence of forces. These are decomposed into *internal* forces, as a result of the expected curve *stiffness*, and *external* forces computed from the image data and distributed over the image. After the snake is initialized inside or outside of an object boundary, it is expected to evolve to this boundary while being controlled by these forces. The internal forces are designed to hold the curve together (*elasticity* forces) and to avoid large bending (*bending* forces). Typically we consider parametric curves and allow them to move toward desired features, usually edges, under the influence of *external* and *internal* forces (Xu and Prince, 1998).

The gesture recognition system can be logically divided into two separate modules: that for hand posture detection and recognition as well as that for dynamic gesture interpretation.

In the past, a robust approach to hand feature extraction and pose classification was proposed (Kasprzak and Skrzyński, 2006), which measured the difference between two active contours. This approach allowed the recognition of 21 poses. One may want to extend this hand set in order to cover a typical alphabet of characters. This can be achieved by allowing a movement of the hand pose in a short image sequence. For dynamic process modeling, stochastic approaches can be chosen, e.g., Hidden Markov Models (HMMs) (Rabiner and Juang, 1993) or (more general) Dynamic Bayesian Networks (DBNs) (Murphy, 2002). The advantage of such techniques is that efficient learning algorithms exist for them (Baum *et al*., 1970; Polanska *et al*., 2006). However, based on the simple "naive Bayes" assumption, the HMM performs well if applied for sentence recognition, even in speech recognition where highly correlated observations appear (Tóth *et al*., 2005). Hidden Markov models have also been applied to the recognition of different gesture languages (e.g., Starner and Pentland, 1995; Kapuściński, 2006).

Thus, our two object recognition modules utilize separate modeling techniques: there is a deterministic approach to hand pose recognition proposed and a stochastic approach, based on hidden Markov models, to hand gesture recognition. In the methods presented so far, the communication between these two modules is strictly unidirectional, namely, the hand pose recognition module provides semi-processed data to the module responsible for gesture recognition. In this paper, we introduce a feedback-like architecture for gesture recognition, by ensuring bi-directional communication between the two recognition modules.

We present a computer vision system for hand gesture recognition that is structured into four steps. We start with basic image segmentation with the goal of hand contour detection (Section 2). Then we use our two active contours approach for hand feature detection and pose classification (Section 3), which is an extended version and a more efficient implementation of our previous approach (Kasprzak and Skrzyński, 2006). The extension of the static hand pose set by "dynamic" signs is possible due to a hand motion detection in a short image sequence (Section 4). Finally, we apply a hybrid model (DBN + hidden Markov models) for pose sequence recognition in a longer image sequence (Section 5). Extensive implementation details and tests verify the validity and quality of the proposed approach (Section 6).

2. Structure of the approach

As usual, we can differentiate between two processing modes of the image analysis system: the **active work** (online) mode and the **system setting** (off-line) mode. The basic steps in the active mode include:

(i) image acquisition and segmentation (Section 3),

(ii) hand pose recognition (single image analysis) (Section 4),

(iii) short-time hand position tracking (short-time trajectory detection) (Section 5),

(iv) long-time gesture recognition (model-based pose sequence recognition) (Section 6).

What is necessary to be done in the system settings mode is

(i) system calibration,

(ii) assigning characters to hand poses (providing the character set),

(iii) learning the hidden Markow model of gestures (acquiring the model of gestures).

The system calibration step (Fig. 1) requires from the user providing a color calibration as well as some parameter values to the system. A white color pattern should be presented to the camera so that the system will adjust the white balance (in normal lighting conditions the white color should be represented in the middle of the color (U,V)-color component scale). After that, the automatic white balance of the camera should be switched off. The system parametrization need basically means informing the system which hand (left or right) will be observed.

The character set is explained in Section 5, while the HMM-based gesture model—in Section 6.

There are some restrictive assumptions about the acquired scene:

- Hand is a foreground object (not occluded), freely located and oriented (no fixed supports or gloves).
- Limited color restrictions posed onto the background.

Fig. 1. Example images in system calibration.

- Natural lighting conditions (no structured light).
- Hand motion speed in accordance with available processing speed.
- The hand sign set is known to the user, who tries to follow this set.

One cannot always expect perfect results of image segmentation and contour classification, but due to the model-based image sequence analysis these eventual low- and intermediate-level errors should largely be eliminated.

3. Image segmentation

In this approach the processing of a single image consists of the following steps: color-based skin region detection, contour initialization, the evolution of two active contours, the creation of a contour difference-based shape description, and hand pose recognition.

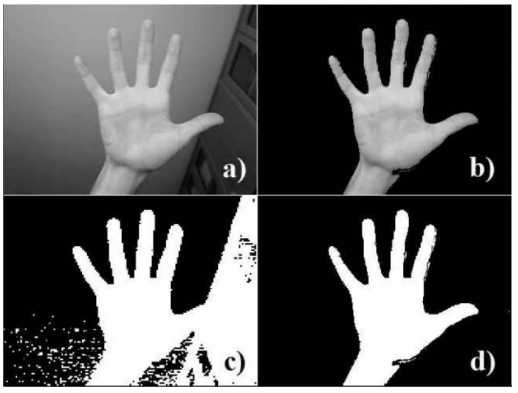


Fig. 2. Image segmentation: original image (a), results of background subtraction (b), results of color recognition (c), results of both background subtraction and color recognition (d).

3.1. Color-based ROI. In the proposed approach the segmentation of the hand image is due to a color-based analysis that leads to the detection of skin colored regions in the image (Kasprzak and Skrzyński, 2006; Wilkowski, 2008). The skin color is detected based on chrominance values, so that small changes in light intensity (e.g., due to shading) do not affect the results of segmentation. In order to handle a complex but static background, background subtraction is used in processing. The main image segmentation steps are illustrated in Fig. 2.

The sensor data are initially given in a 24-bit RGB color scheme. It is evident that this color space is not well designed for color-based object recognition. Among different alternative color spaces, we consider the YUV and YCbCr spaces to be most suitable for this task, as the explicit intensity component Y allows creating a particular color normalization scheme. Let us take, for example, the YUV space. The Y component represents brightness, while U and V are computed as scaled differences between Y and the B and R values, respectively. Transformation from the RGB to the YUV space (with components represented by 8-bit unsigned integers, i.e., values from the interval $[0, 255]$) may be performed as follows:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (1)$$

We applied a normalization of the U and V components, driven by the normalization of Y to a predefined value (let us fix it to 128). Due to this procedure we can achieve very stable and narrow intervals, centered at 112 (for U) and 149 (for V), which represent the skin color. Observe that in a color calibration table (called *Ideal Mackbeth* YUV) the "dark skin" and "light skin" colors are represented by very different (YUV)-values: $(88, 114, 146)$ and $(161, 110, 151)$, respectively. The normalized components are computed as

$$\begin{bmatrix} Y_{\text{norm}} \\ U_{\text{norm}} \\ V_{\text{norm}} \end{bmatrix} = \begin{bmatrix} 128 \\ \kappa \cdot (U - 128) + 128 \\ \kappa \cdot (V - 128) + 128 \end{bmatrix}, \quad (2)$$

$$\kappa = \begin{cases} \eta \dfrac{Y_{\text{norm}}}{Y} & \text{for} \quad Y > Y_{\text{norm}}, \\ \theta \dfrac{256 - Y}{Y_{\text{norm}}} & \text{for} \quad Y < Y_{\text{norm}}, \\ 1 & \text{for} \quad Y = Y_{\text{norm}}, \end{cases} \quad (3)$$

In particular, after setting $\eta = 1.135$ as well as $\theta = 1/\eta \simeq 0.881$, the "dark" and "light" skin colors are normalized to a common triplet $(Y_{\text{norm}}, U_{\text{norm}}, V_{\text{norm}}) = (128, 111.8, 148.8)$.

Without the intensity-based normalization, color-based skin detection may sometimes fail, especially in difficult lighting conditions. If shadows or strong lighting fall onto the shin surface, then the colors significantly change.

Also, in the case of a very structured dynamic background containing colors similar to the skin color, palm detection problems may arise. Obviously, other color correction solutions may be tried, like, for example, luminance regularization. We have applied image smoothing to an image that contained the detected skin pixels.

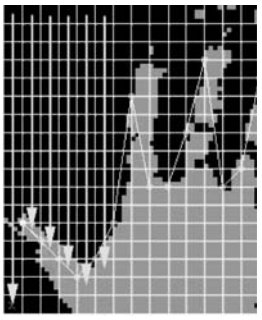


Fig. 3. Initial contour detection.

3.2. Contour initialization. The input data to this step consist of the binary image containing skin-color pixels. The binary image is divided into 3×3 or 4×4 blocks (the block size is a parameter). First, the blocks are examined from left to right and from top to down. If the block examined contains a sufficient amount of skin pixels (let us call it the "block filling" threshold), then the next active contour point is located in the left upper corner of this block. In such a case, the search moves to the next column on the right side. This cycle is iterated until the end of the image is reached or a no-skin column is found (see Fig. 3). In this way the upper part of the initial contour is located. Next, the blocks are examined from right to left and from bottom to up to locate the bottom part of the initial contour. The control point of the active contour is now situated in the bottom right corner of each positively verified block found.

4. Two active contours

In this section we are going to demonstrate how the double-contour approach allows us to recognize 21 different static poses. This variability is achieved due to simple features, like the general type of hand view, the number of visible fingers and the location of the thumb.

4.1. Active contour. Snakes (active contours) are curves defined in the image plane that can change their shape and move under the influence of forces. These are decomposed into *internal* forces, as a result of the expected curve *stiffness*, and *external* forces computed from the image data and distributed over the image. After the snake is initialized inside or outside an object boundary, it is expected to evolve to this boundary while being controlled by these forces. The internal forces are designed to hold the curve together (*elasticity* forces) and to avoid a large bending (*bending* forces).

4.2. Balance of forces. We consider parametric curves and allow them to move toward desired features, usually edges, under the influence of *potential* forces, which are defined to be the negative gradient of a potential function. The active contour is defined in the image plane as a sequence of control points $p_i = (x_i, y_i)$, $(i = 0, \ldots, n-1)$ and connecting line segments $L_i = [p_i, p_{i+1}]$. During contour evolution the total energy of points is minimized.

The external energy E_{ext} is traditionally estimated as a potential energy (the sum of image function $I(x, y)$) measured in control points:

$$E_{\text{ext}} = K_0 \cdot \sum_{i=0}^{n-1} I(x_i, y_i), \tag{4}$$

where K_0 is a scaling coefficient. The appropriate external force is computed as the negative gradient of this potential energy function ($F_{\text{ext}} = -\nabla E_{\text{ext}}$). It pulls the contour toward the desired image edges.

The internal energy E_{int} consists of two components:

$$E_{\text{int}} = E_{\text{elastic}} + E_{\text{stiffness}}. \tag{5}$$

The *elasticity* energy is proportional to the squared contour length:

$$E_{\text{elastic}} = K_1 \cdot \sum_{i=0}^{n-1} |p_i - p_{i-1}|^2, \tag{6}$$

where K_1 is an elasticity coefficient and the p_i s are the contour's control points. The *elasticity* force tends to shorten the line segments, i.e., to move the control points closer to their neighbors.

The *stiffness* energy is proportional to the squared curvature measured at control points:

$$E_{\text{stiffness}} = K_2 \cdot \sum_{i=0}^{n-1} |p_{i-1} - 2p_i + p_{i+1}|^2, \tag{7}$$

where K_2 is a stiffness coefficient. The *stiffness* force tends to lower the curvature, i.e., it avoids bending the contour without shortening it.

The appropriate elasticity and stiffness forces can be derived by using the energy gradient (i.e., $\mathbf{F} = -\nabla E$). In the final stable state of the contour a balance of forces for a given contour in a given image is achieved:

$$\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}} = 0. \tag{8}$$

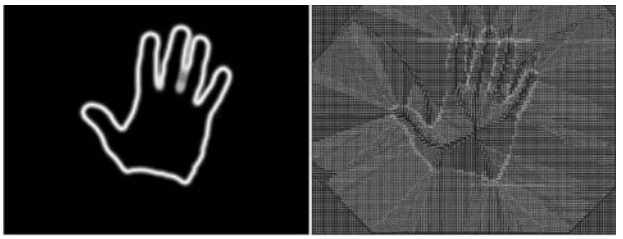


Fig. 4. Gradient image (left) and the gradient vector field (with $m = 0.2$) (right) computed from a hand image with a uniform background.

4.3. Contour update rule. Based on the initial border of the most probable hand region, the active contour method is run twice with different parameter settings. In the basic iteration loop, for every contour point $p_i = [x_i, y_i]$ we apply the update equations:

$$x_i = x_i + a \cdot F_{\text{elastic}}(X, i) + b \cdot F_{\text{stiff}}(X, i) + g \cdot F_{\text{ext}}(X, i), \quad (9)$$

$$y_i = y_i + a \cdot F_{\text{elastic}}(Y, i) \quad (10)$$

$$+ b \cdot F_{\text{stiff}}(Y, i) + g \cdot F_{\text{ext}}(Y, i), \quad (11)$$

where a, b, g are some weight parameters and $F(X, i)$ or $F(Y, i)$ denote the force component along the X axis or the Y axis, respectively, measured at p_i.

4.4. Outer hand and inner palm contours. In a previous paper, we applied another approach for the computation of external forces, called the *Gradient Vector Field* (GVF) (Xu and Prince, 1998). This allowed better moving the snake into boundary concavities. The GVF is a dense vector field, $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$, derived from the image by minimizing a certain energy functional in a variational framework (Fig. 4). The minimization of the functional is a highly computationally expensive solution and unsuitable for an on-line tracking mode of the image analysis system. It was observed, however, that retrieval of the precise shape of the hand is usually not crucial for recognition of hand posture (e.g., the number of fingers can be successfully counted even if concavities between fingers are not exactly filled by the contour snake). For these two reasons the computation of the GVF was dropped in our final solution.

By varying the internal "force weights" in the active contour method these iterations converge to two different contours: (i) one covering both the palm and fingers, and (ii) one related to the palm only.

For detection of the first (outer) contour, the parameters were set to $a = 0.1, b = 0.1, g = 0.6$.

The second contour should detect the palm area of the hand only, i.e., it should not include the fingers. For this purpose the parameter g is set to 0.15. After this change the *elasticity* force is more dominant than in the first case, and this leads to a contour shorter than the first one.

The setting of parameter a allows controlling the smoothness of the outer contour. With a higher value of a (e.g., 0.8), the approach converges more slowly and imperfectly reaches the finger boundaries in comparison to the case of $a = 0.1$ (Fig. 5). With a lower value of a the influence of the internal forces is small and the contour can approach exactly the image edges.

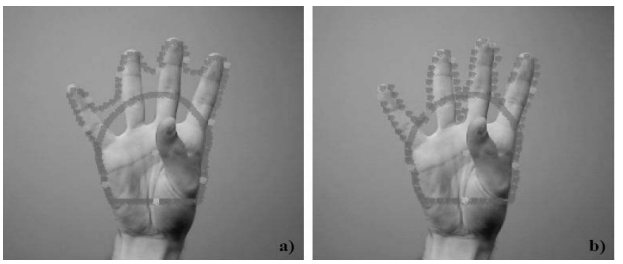


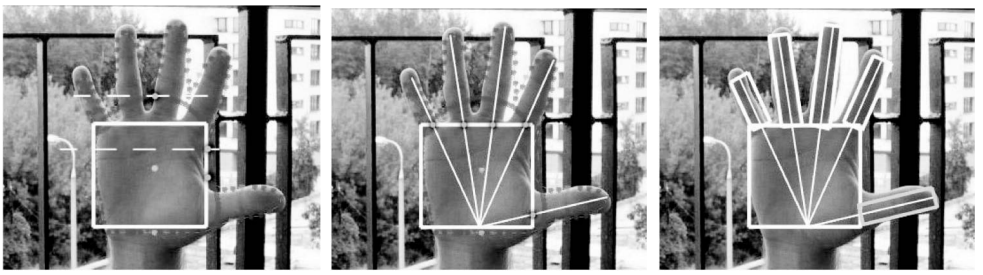
Fig. 5. Two different outer contours related to the hand achieved for $a = 0.8$ (a) or $a = 0.1$ (b).

4.5. Contour difference description. A relocation of the contour's control points is established due to an approximation of existing points. The new control points are uniformly distributed and their number is fixed. Next the center of mass of the inner contour is determined. The crossings of lines $y = x + c_1$ and $y = -x + c_2$ that goes through the central point with the bottom part of the inner contour allows elimination of its forearm part.

The distances of all contour points from this mass are computed, starting from the point that is located lowest in the image and nearest to the image column in which the center of mass is located (Fig. 6). Thus we obtain two 1-D distance distributions. These distributions are subtracted one from the other and length-normalized, and all negative values are reset to zero. In the resulting function the visible tips of fingers correspond to local maximum points (Fig. 7).

Having computed the contour difference description we can proceed to finger and palm detection.

The current hand pose is generated from the hand contour description in three steps. The center of mass and other specific boundary points of the inner contour determine the location and size of the palm (Fig. 8(a)). The fingers are detected based on the locations of finger tips with respect to the projection point of the center of mass onto the palm rectangle side, opposite to the tips (Fig. 8(b)). The finger is detected as a thumb when its tip projection onto the palm rectangle is located below the mass center point (Fig. 8(c)).



(a) (b) (c)

Fig. 8. Detected hand pose: palm detection (a), finger detection (b), thumb detection (c).

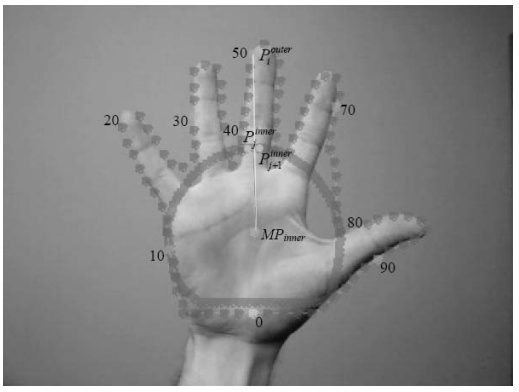


Fig. 6. Computing differences between corresponding points of the two contours. The correspondence is established with respect to the mass center of the inner contour MP_{inner}.

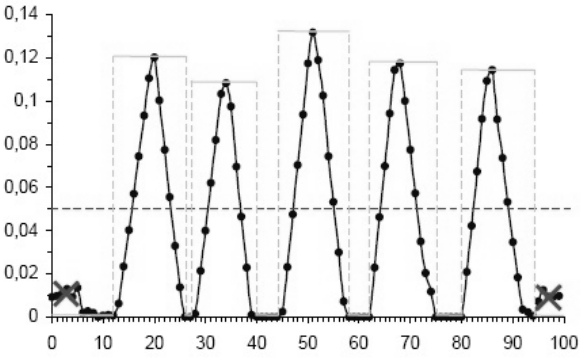


Fig. 7. Local maxima of the difference function.

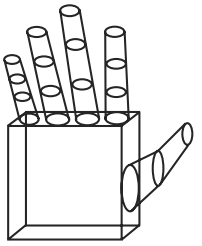


Fig. 9. Simple 3-D palm model.

4.6. Hand pose classification. Finally, a hand pose classification based two contour descriptions can be performed and this pose can then be assigned a specific character from a set of 21 characters. The implicit hand model consists of a box representing the palm and five cylinders that represent the 5 fingers (Fig. 9). The relation of the palm rectangle width to outer-contour-height induces the front or side view, while the height-to-height relation for outer and inner contour allows differentiating between frontal open palm and fist postures (Fig. 10). In order to classify the characters, the following features are utilized: the position of the thumb, the number and configuration of fingers and hand view (hand orientation) (see Tabela 1).

5. Hand pose set

5.1. Basic character set. The simple but quite reliable set of double contour-difference features allows detection of 21 static hand poses. One pose will be reserved for the *break* sign, which will terminate every gesture. The other poses were assigned 20 characters of the Polish alphabet that appear most often in the language (Przepiórkowski, 2006) (Fig. 11).

Nine diacritic characters (marked by a boundary box) are omitted, first. There are three letters, marked by a stair, (Q, V, X), which are both rare and of low importance. For example, they do not appear in the Polish finger alphabet PAP. We eliminate them from consideration. Finally, Fig. 12 illustrates the 20 characters assigned to static hand poses. The letter 'J' has meaning similar to 'I', but appears relatively seldom. Hence this letter will be represented by a "dynamic" version of the pose for letter 'I'. The characters 'G' and 'H' will be dynamic versions of 'W' and 'Y', appropriately. The termination sign corresponds to a "closed" fist pose (Fig. 13).

5.2. Hand motion detection. By detecting a hand motion in consecutive several images, we change the meaning of some characters, to be compatible with the thumb alphabet (Fig. 14). This short-time tracking of hand pose

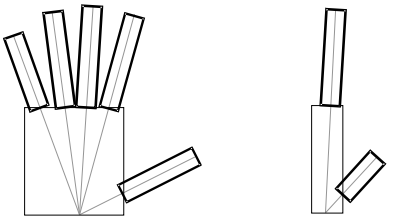


Fig. 10. Front- and side-views of the palm model.

A	8,91%	W	4,65%	P	3,13%	G	1,42%	Ć	0,40%
I	8,21%	S	4,32%	M	2,80%	Ę	1,11%	F	0,30%
O	7,75%	T	3,98%	U	2,50%	H	1,08%	Ń	0,20%
E	7,66%	C	3,96%	J	2,28%	Ą	0,99%	Q	0,14%
Z	5,64%	Y	3,76%	L	2,10%	Ó	0,85%	Ź	0,06%
N	5,52%	K	3,51%	Ł	1,82%	Ż	0,83%	V	0,04%
R	4,69%	D	3,25%	B	1,47%	Ś	0,66%	X	0,02%

Fig. 11. Expected frequencies of characters in the Polish language.

is an intermediate processing step between single image pose classification and gesture recognition. The motion is computed as a time displacement of the central point of the hand. The motion computed is quantized into five distinct values: no-motion and motion in each of four major directions.

By considering the motion information for the basic hand pose we can generate modified poses, which usually corresponds to diacritic characters, like 'Ą', 'Ć', 'Ę', 'Ł', 'Ń', 'Ó', 'Ś', 'Ż' (Fig. 15). Additionally, the remaining characters from the alphabet are detected, which were omitted in the first single-image processing stage (e.g., the "moving" version of 'I' is 'J' and, similarly, 'W' → 'G', 'Y' → 'H') (Fig. 16).

6. Gesture recognition by bi-driven inference

6.1. Gesture representation. A *hidden Markov model*, HMM = $(S, C, \Pi, \mathbf{A}, \mathbf{B})$, represents a stochastic process in time, in terms of (hidden) states S, (visible) observations C, initial state probabilities Π, state transition probabilities A and output probabilities B (Rabiner and Juang, 1993). Its special case, the left-right HMM, is useful to represent possible state paths that correspond to observation sequences (Kasprzak, 2009) (Fig. 17).

The model is designed in two stages. First, the number of states and the model structure have to be determined. For every gesture, its own left-right sub-model is created (Fig. 18). The number of HMM states matches the number of different poses which constitute the given gesture. Then the model parameters need to be trained: $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$. The *Baum–Welch training* is applied (Baum *et al.*, 1970), which is a Maximum Likelihood (ML) procedure that iterates over the learning set and updates this parameter set in order the maximize the prior probability:

$$\sum_{o_i} \log \mathbf{P}(o_i|\lambda_j), \tag{12}$$

where $o_i = o_i^1 \dots o_i^{T_i}$ is the i-th training sequence for the j-th model and λ_j is the set of the j-th model parameters.

Finally, the sub-models are integrated into a single model, by connecting their "initial" states with a common start state and, similarly, their "final" states with a common terminal state.

The goal of *Viterbi search* is to find a path in the model (leading from the start state to the terminal state) that has the maximum posterior probability given the current observation sequence:

$$\begin{aligned}&\mathbf{P}(S^1 S^2 \dots S^t S^{t+1} | o^1 o^2 \dots o^t o^{t+1}) \\ &= \mathbf{P}(o^{t+1}|S^{t+1}) \\ &\quad \cdot \max_{S^{t+1}} [\mathbf{P}(S^{t+1}|S^t)\mathbf{P}(S^1 \dots S^t | o^1 \dots o^t)],\end{aligned} \tag{13}$$

where $S^1 S^2 \dots S^t S^{t+1}$ is the sequence of hidden states in the HMM, and $o^1 \dots o^t$ denotes the corresponding obse-

Table 1. Set of 21 signs and their codes corresponding to hand instances.

No.	Thumb position	Detected fingers	Hand view	Character
1	Not	closed (0)	Front/Back	break
2	Not	5 together (0)	Front/Back	O
3	Not	1	Front/Back	D
4	Not	2	Front/Back	Y
5	Not	3	Front/Back	W
6	Not	4	Front/Back	N
7	Not	5 together (0,1)	Side	I
8	Left	thumb (1)	Front/Back	R
9	Left	4 together + thumb	Front/Back	B
10	Left	2	Front/Back	L
11	Left	3	Front/Back	P
12	Left	4	Front/Back	F
13	Left	5	Front/Back	M
14	Left	4 together + thumb	Side	U
15	Right	thumb (1)	Front/Back	A
16	Right	4 together + thumb	Front/Back	Z
17	Right	2	Front/Back	C
18	Right	3	Front/Back	E
19	Right	4	Front/Back	S
20	Right	5	Front/Back	T
21	Right	4 together + thumb	Side	K

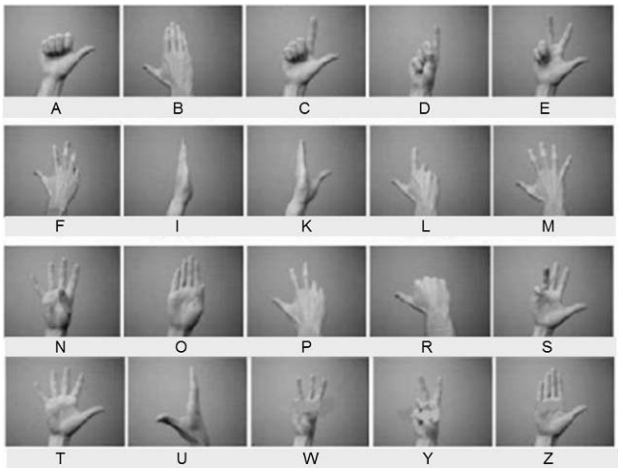


Fig. 12. Twenty characters assigned to static poses.

Fig. 13. Word termination sign.

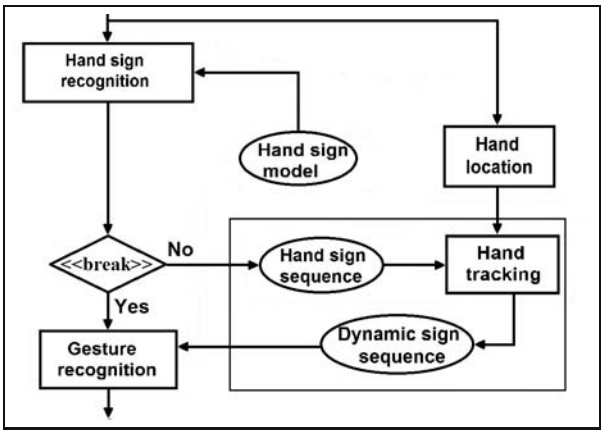


Fig. 14. Structure of the short-time tracking step.

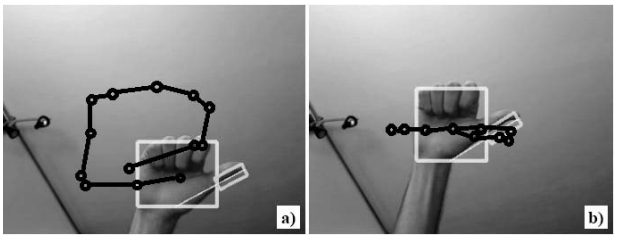


Fig. 15. Paths of the hand contour center detected in an image sequence: random path (a), letter 'Ą' represented by a path and constant pose 'A' (b).

rvation sequence. In our approach the observation sequences are constructed from vectors representing similarities between the hand detected and each of recognized symbols. The similarity measure is computed from the similarity of particular features such as hand orientation, the number of fingers visible, and the orientation of a thumb.

6.2. Bi-driven inference. We have designed a hybrid system for gesture recognition, incorporating both recognition modules (pose and gesture recognition) into one scheme and ensuring bi-directional communication between them.

The benefits of bi-directional communication are obvious: the top layer module which possesses the information on the structure of gesture recognized can provide the bottom layer module with the data permitting to accurately predict the most expected hand shape and position in order to increase the accuracy of hand detection. This concept has been widely used in the problems of object tracking such as Kalman filters or particle filters; however, what we propose here is its extension to the task of gesture recognition which comprises both the tracking of the hand through its different shape, orientation and position configurations, but also simultaneous recognition of the gesture performed.

In our view the most convenient method for modeling the dynamics of this hybrid system is the dynamic Bayesian network stochastic model (Murphy, 2002). This model ensures maximum flexibility in defining relationships between different system parameters while providing general inference and learning techniques. An additional advantage is that the probabilistic properties of the model enable to easily represent uncertainty of the recognition process. The most general form of the network that is proposed is given in Fig. 19.

The main probabilistic element in the network is given as S_t. This is actually composed of several different discrete variables that together describe the current system state:

- current probabilities of words W_t, i.e., the "observation" variable corresponds to situations of HMM-based word recognition processes for every possible word;
- the mode of hand expression dynamics (variable D_t); could also be interpreted as the stage of single word recognition, and intuitively associated with the HMM hidden state;
- current hand posture probabilities (variable H_t) comprise posture "observations" that depend on both on

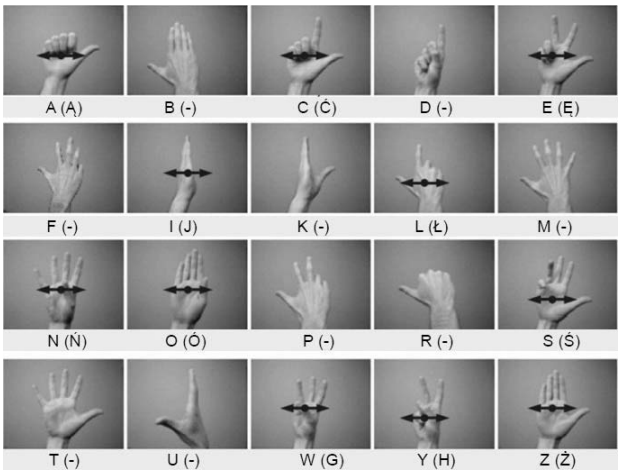


Fig. 16. Characters assigned to dynamic poses.

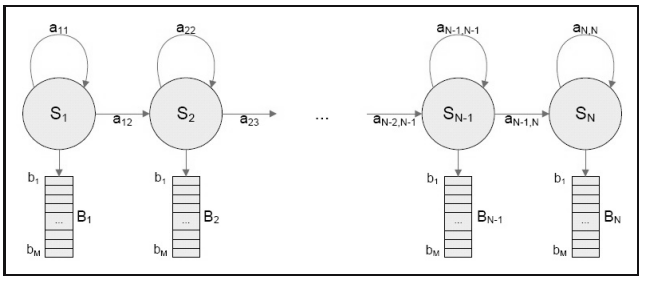


Fig. 17. Left-right hidden Markov model.

states W_t and D_t; this state is also used as a "selector" of the motion model for the given stage of a gesture.

In order to simplify transition probabilities, another variable denoting reaching the final state of word recognition (F_t) can be used as well. The relationships between particular component variables constituting the meta-variable S_t can be further rewritten in the form of a separate network. The bottom-up dependence needed for a recursive state filtering process allows the current posterior probability estimations of competitive words and hand dynamics. With this information, these two variables control (modify), in a top-down manner, the probabilistic distribution of a stochastic variable, created for "hand posture" representation.

The continuous variable X_t represents the dynamical parameters of the hand which may comprise its position, velocity and acceleration along different axes and dimensions (e.g., xy-coordinates, the rotation angle). The value of the variable depends both on its value in the previous time-step as well as on the value of the discrete control variable S_t (in particular H_t). The latter relationship makes the system "trajectory-aware" since each phase of the gesture performed can have a different model of hand motion associated.

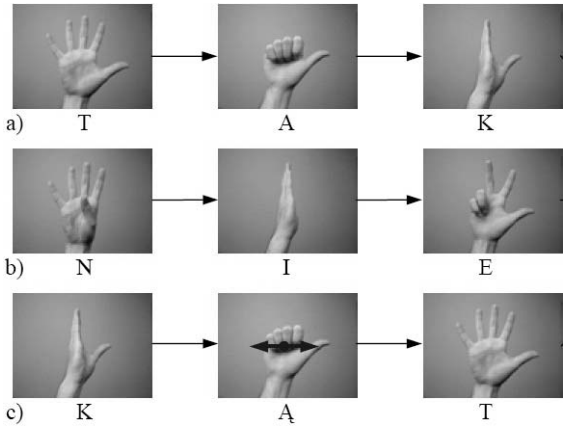


Fig. 18. Words as sequences of hand poses: 'TAK' (yes) (a), 'NIE' (no) (b), 'KĄT' (angle) (c).

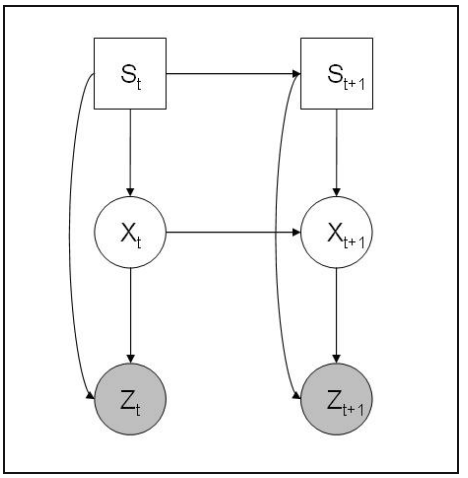


Fig. 19. Proposed Bayesian network: iterative structure of two-slices of temporal representation.

Z_t is the observed variable that represents the measurable features of the hand that are retrieved from the image. This variable depends on both the value of the discrete variable H_t (describing the current hand posture) and the subset of parameters from the continuous variable X_t (in particular hand position, size and orientation).

The experiments for this proposed control model used two inference methods common for DBNs: a particle filter (Arulampalam *et al.*, 2002) and a switching Kalman filter (Murphy, 1998). The main critical issue identified was the efficiency trade-offs resulting from the need for tracking multiple hypotheses (i.e., extending words in a letter-per-letter manner) while using these two inference methods. This problem becomes especially apparent when using larger vocabularies of available hand postures and gestures.

7. Implementation and experiments

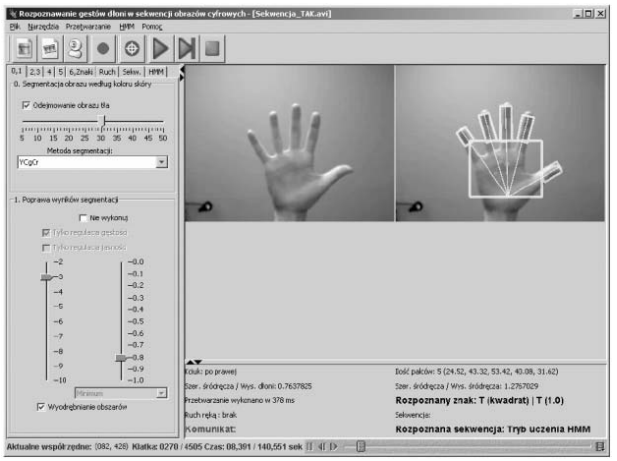


Fig. 20. Testing environment.

7.1. Implementation. The system was implemented as a desktop application in the Java language (Fig. 20). The image sequences were acquired by a low cost digital camera Logitech QuickCam Pro 9000, with a sensor matrix resolution of 1600×1200 pixels and the following settings: automatic white balance, autofocus and automatic exposure time; however, the automatic white balance is kept turned off in the phase of recognition. To satisfy the on-line processing conditions, a low-resolution image sequence was acquired frames with the resolution of 320×240 pixels, 24 bits RGB pixel and 30 frames per second.

7.2. Experiments. Some constraints were posed on the lighting conditions—although either natural or synthetic lighting were applied, highlighting and dark conditions have been avoided. The scene was constrained, too, to contain only one human skin region, e.g., a single hand and no face, etc. There could be a nonhomogeneous background that contains only small skin-color elements unless they are part of the static background and can be removed by background subtraction.

The recognition quality is expressed by the percentage of properly recognized signs or gestures related to the total number of analyzed frames or sequences.

The training set consisted of 160 test sequences for 5 gestures (words), i.e., 32 sequences per word each (located in 40 AVI files). These gestures corresponded to following words: ‘TAK’, ‘NIE’, ‘KĄT’, ‘NIC’, ‘PIES’. In the testing stage other 160 sequences, acquired on-line, were recognized. It appeared that the system works nearly perfectly if the poses are accurately shown, otherwise some misclassifications of single poses are possible (Fig. 21). In the left part of the figure, the fingers are not properly separated from each other. The inner contour did not sufficiently “deeply” move between the fingers, leading to insignificant local maxima of the two-contour-difference function. In the right part of the figure (b) the thumb is inside the palm area, thus leading to a misinterpretation of the thumb and the small finger.

Obviously, the number of gestures was small compared with the potential total number of gestures that can be represented by 4-character words ($= 30^4$). Many other gestures could be wrongly accepted if the recognition quality was considered. For this reason, a “do not know” answer was added to the gesture recognition stage. If the best solution sentence has a relatively low quality score, then a “no recognition” result is reported.

We studied only a small set of short-length gestures, but where the individual characters repeatedly appear, like: ‘N’ in ‘NIE’ and ‘NIC’, ‘K’ in ‘KĄT’ and ‘TAK’, or ‘I’ in ‘NIE’, ‘NIC’ and ‘PIES’. Hence, these gestures were not far apart from each other in the large coding space, and so this situation is mimicking the conditions of a much larger gesture set. Only 10 static postures appear in these sentences. However, the selection of postures used in the experiments implies that the use of a modified set (providing that hand posture order in the sequence allows smooth transitions during the gesture, making the gesture easy to perform) should give comparable results.

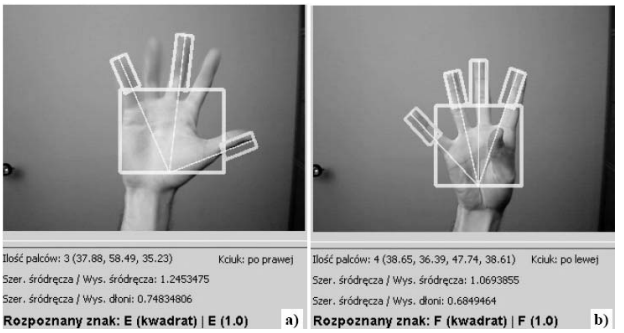


Fig. 21. Examples of single pose errors.

P P P P P P P I I I I I **D** E E E **C** E E S S S S S S S S S S S
P P P P P P P I I I I I **D** E E E E E E **C** S S S S S S
P P P P **L** P P P I I I I I I I **D** E E E E E E E E E E S S S S S S
P P P P P P P I I I I I I **D** E E **C** E E S S S S S S S S S S S S **E**
P P P P P P P **L** I I I I E E E E E E E **C** S S S S S S
P P P P **L** P P P I I I I I I E E E E E E **C** S S S S S S S
P P P P P P P I I I I I E E E E E E **C** S S S S S S S
P P P P P P **L** I I I I I **D** E E E E E E **C** S S S S S S S

Fig. 22. Examples of measured letter sequences for the word ‘PIES’ (‘dog’). Every row represents one sequence. The letter errors are given in bold. All the words are properly recognized.

The worst recognition rate of single characters (poses) was from around 75% (for such letters as ‘K’, ‘T’, ‘O’, ‘D’, *break*) to around 90% for the best recognized

letters (e.g., 'N', 'I', 'E', 'P', 'S', 'A', 'C'). To a large extent these single letter errors were compensated by the final gesture recognition stage as the word recognition rate reached nearly 95% (Fig. 22).

The frequent source of hand pose misclassifications was confusing the "side" and "front" views of the palm in the case of side-like postures with a significant side profile width. In this way the character 'K' was sometimes falsely classified as the character 'C' or 'Z'. Other errors originated from mistaking the frontal hand pose with all five fingers together (like the character 'O') with the frontal hand position with a single finger visible (the character 'D') or all fingers folded (the "break" sign). A similar mechanism was also responsible for taking the "break" sign for the characters 'D', 'A' or 'Z'.

Mistaking frontal and side hand pose or detection of additional fingers were due to the errors in fitting of the inner contour, while other inaccuracies in the counting fingers were the result of an improper fit of the outer contour (e.g., 'T' was mistaken for 'S', 'N', 'W' or 'E').

Another source of errors was unknown transient postures that appear at the moment of transition between known hand postures. Some problems were also due to an inaccurate hand orientation with respect to the model orientation (e.g., tilted hand).

Since the "break" sign is fundamental in temporal segmentation of gestures, it received a special treatment in our system. In order to make the system robust to occasional misclassifications, we imposed a condition concerning the ratio of appearance of the "break" sign in a subsequence required to make the decision concerning word ending.

The system proved to be robust to different lighting and background conditions. The system achieved comparable results both in natural and artificial light. Also, a complex but static background (including hand-color-like objects) was well handled due to background subtraction. Most of the problems were observed when the hand was artificially back-lit.

The processing time of a single frame depends on the content—on our PC with Sempron 3000+, 1.8 GHz, the processing times were within the range from 200 ms (letter 'I') to 400 ms (letter 'T'). The gesture recognition stage with 5 words in our dictionary needs only a few ms per frame.

8. Summary

In this paper our previous approach to hand sign recognition (Kasprzak and Skrzyński, 2006) was modified to handle hand pose sequences with the speed of several frames per second on a typical PC. A trade-off between single pose recognition quality and computational speed was achieved. At the price of increased single-pose errors, we managed to speed up the application to work in real-time. Much effort was spent reliably on recognizing the "break" sign. The drawback of introducing larger pose classification errors is compensated with the help of context information induced by the gesture model. The modules for short-time hand tracking and for gesture recognition were added. A bi-driven control of the word and letter-recognition stages was designed. The proposed approach is quite universal, and different letters and words can be assigned to hand poses and gestures, depending on the application domain.

Acknowledgment

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

References

Arulampalam, M.S., Maskell, S. and Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing* **50**(2): 174–188.

Baum, L., Petrie, T., Soules, G. and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annal Mathematics Statistics* **41**(1): 164–171.

Emambakhsh, M., Ebrahimnezhad, H. and Sedaaghi, M.H. (2010). Integrated region-based segmentation using color components and texture features with prior shape knowledge, *International Journal of Applied Mathematics and Computer Science* **20**(4): 711–726, DOI: 10.2478/v10006-010-0054-y.

Flasiński, M. and Myśliński, S. (2010). On the use of graph parsing for recognition of isolated hand postures of Polish sign language, *Pattern Recognition* **43**(6): 2249–2264.

Fu, C.-S., Cho, W. and Essig, S. (2000). Hierarchical colour image region segmentation for content-based image retrieval system, *IEEE Transactions on Image Processing* **9**(1): 156–162.

Gonzalez, R.C. and Wintz, P. (1987). *Digital Image Processing*, Addison-Wesley, Reading, MA.

Kapuściński, T. (2006). *The Recognition of the Polish Sign Language in a Vision System*, Ph.D. thesis, University of Zielona Góra, Zielona Góra, (in Polish).

Kasprzak, W. (2009). *Image and Speech Signal Recognition*, WUT Press, Warsaw, (in Polish).

Kasprzak, W. and Skrzyński, P. (2006). Hand image interpretation based on double active contour tracking, *in* T. Zielińska and C. Zieliński (Eds.), *ROMANSY 16. Robot Design, Dynamics, and Control*, CISM Courses and Lectures, Vol. 487, Springer, Wien/New York, NY, pp. 439–446.

Kass, M., Witkin, A. and Terzopoulos, D. (1998). Snakes. Active contour models, *International Journal of Computer Vision* **1**(4): 321–331.

Marnik, J. (2003). The recognition of characters from the Polish finger alphabet, *Technical report*, StatSoft Polska, Cracow, `http://www.statsoft.pl/czytelnia`

/badanianaukowe/d0ogol/marnik.pdf, (in Polish).

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley, CA.

Murphy, K.P. (1998). Switching Kalman filters, *Technical report*, DEC/Compaq Cambridge Research Labs, Cambridge, MA, http://www.cs.berkeley.edu/~murphyk/Articles/skf.ps.gz.

Niemann, H. (2000). *Klassifikation von Mustern*, Springer, Berlin.

Pitas, I. (2000). *Digital Image Processing Algorithms and Applications*, Prentice Hall, New York, NY.

Polanska, J., Borys, D. and Polanska, A. (2006). Node assignment problem in Bayesian networks, *International Journal of Applied Mathematics and Computer Science* **16**(2): 233–240.

Przepiórkowski, A. (2006). Frequency of letters in written Polish, Linguistic Advisory Website of Polish Scientific Publishers (PWN), http://poradnia.pwn.pl/lista.php?id=7072.

Rabiner, L. and Juang, B. (1993). *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ.

Rafajłowicz, E., Wnuk, M. and Rafajłowicz, W. (2008). Local detection of defects from image sequences, *International Journal of Applied Mathematics and Computer Science* **18**(4): 581–592, DOI: DOI: 10.2478/v10006-008-0051-6.

Rehg, J. and Kanade, T. (1993). Digit eyes: Vision-based human hand tracking, *Technical Report CMU-CS-93-220*, School of Computer Science, Carnegie Mellon University, Pittsburg, PA.

Sanchez-Reillo, R., Sanchez-Avila, C. and Gonzalez-Marcos, A. (2000). Biometric identification through hand geometry measurements, *Transactions on Pattern Analysis and Machine Intelligence* **22**(10): 1168–1171.

Starner, T. and Pentland, A. (1995). Visual recognition of American sign language using hidden Markov models, *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, Zurich, Switzerland*, pp. 189–194.

Terzopoulos, D. (2003). Deformable models: Classic, topology-adaptive and generalized formulations, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, Springer-Verlag, New York, NY, pp. 21–40.

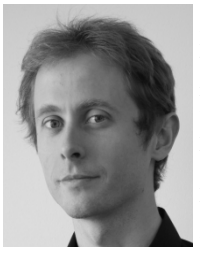
Tóth, L., Kocsor, A. and Csirik, J. (2005). On naive Bayes in speech recognition, *International Journal of Applied Mathematics and Computer Science* **15**(2): 287–294.

Wilkowski, A. (2008). An efficient system for continuous hand posture recognition in video sequences, *in* L. Rutkowski, R. Tadeusiewicz, L. Zadeh and J. Zurada (Eds.), *Computational Intelligence: Methods and Applications*, EXIT, Warsaw, pp. 411–422.

Xu, C.-Y. and Prince, J. (1998). Snakes, shapes, and gradient vector flow, *IEEE Transactions on Image Processing* **7**(3): 359–369.

Yining, D., Manjunath, B. and Shin, H. (1999). Colour image segmentation, *Computer Vision and Pattern Recognition, IEEE Computer Society Conference, CVPR'99, Fort Collins, CO, USA*, Vol. 2, pp. 2446–2451.

Włodzimierz Kasprzak received a Ph.D. in computer science (1987) and a D.Sc. in automatic control (2002) from the Warsaw University of Technology (WUT), and a Dr.-Ing. in pattern recognition (1996) from the University of Erlangen–Nuremberg, Germany. Since 1997 he has been with the Institute of Control and Computation Engineering at the WUT, currently appointed as an associate professor. His main interests are in pattern recognition and artificial intelligence, and their applications in image and speech analysis. Among others, he has conducted research at the University of Erlangen–Nuremberg, the Bavarian Research Center FORWISS, Germany, and the RIKEN Institute, Japan. He is a member of the IAPR.



Artur Wilkowski received the Dipl.Eng. and M.Sc. degrees in applied informatics from the Warsaw University of Technology in 2004. Currently he is pursuing his Ph.D. degree in computer science at the Institute of Control and Computer Science, and is a research and teaching associate at the Faculty of Geodesy and Cartography, Warsaw University of Technology. His research interests include vision-based computer interaction and probabilistic modeling of dynamic systems.

Karol Czapnik received the M.Sc. degree in computer science and information engineering in 2009 from the Warsaw University of Technology, Poland. He is currently active as an iOS developer, especially with regard to mobile technology solutions. His professional interests are in digital imaging, computer vision and the Global Positioning System (GPS).

Received: 10 December 2010
Revised: 26 May 2011
Re-revised: 12 September 2011