Hand Gesture Recognition in Image Sequences Using Active Contours and HMMs

Włodzimierz Kasprzak, Artur Wilkowski, and Karol Czapnik

Institute of Control and Computation Engineering
Warsaw University of Technology
ul.Nowowiejska 15/19, 00-665 Warszawa, Poland
W.Kasprzak@ia.pw.edu.pl, A.Wilkowski@ia.pw.edu.pl

Summary. The created vision system captures image sequences from the digital camera and it first detects static hand poses in every single frame due to a double-active contour classification. The tracking of the hand pose in a short sequence allows to detect "modified poses", like diacritic letters of polish alphabet. Finally, by tracking hand poses in a longer image sequence, this pose sequence ic classified in terms of gestures (words). Hidden Markov Models and Viterbi search are applied for word modelling and recognition at this stage.

1 Introduction

Current research on gesture recognition in image sequences concentrates on the flexibility of system's use, i.e. "free hands" without gloves on a freely structured background, and on the general-purpose architecture of the recognition system, i.e. to find declarative languages for model representation, learning and recognition. The main motivation of such research is to make the man-machine interface more flexible and more easy for the user.

The "free" image-based gesture recognition is decomposed into three main stages: single frame preprocessing and segmentation, hand feature extraction and pose classification, and hand tracking and pose sequence interpretation. Static hand poses with some limited movements have been studied from point of view of the "polish finger alphabet" (PAP) [6], [2]. The Author's approach to feature extraction and pose classification [3] concentrates around the detection of two active contours [5], [11]. For dynamic process modelling stochastic approaches can be chosen, e.g., Hidden Markov Models (HMMs) [9] or Dynamic Bayesian Networks [7].

In this paper we extend our previously recognized pose set, by tracking hand motion in a short image sequence (section 2) and propose the use of HMMs for pose sequence recognition in longer image sequences (section 3).

2 Hand pose detection

2.1 Image segmentation

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 [3], [10]. The skin color is detected on base of chrominance values, so that small changes in light intensity (e.g. due to shading) does not affect the results of segmentation (Fig. 1).

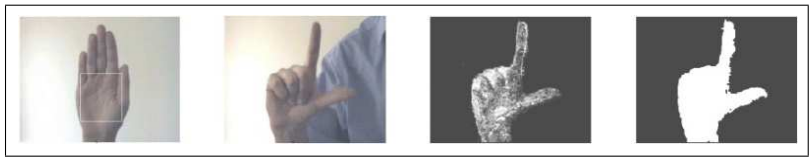

Fig. 1. The steps in human skin detection: (from left to right) skin color calibration, processed frame, pixel classification, binary image.

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

$$x_i = x_i + a \cdot F_{\text{elastic}}(X,i) + b \cdot F_{\text{stiff}}(X,i) + g \cdot F_{\text{ext}}(X,i) \quad (1)$$

$$y_i = y_i + a \cdot F_{\text{elastic}}(Y,i) + b \cdot F_{\text{stiff}}(Y,i) + g \cdot F_{\text{ext}}(Y,i) \quad (2)$$

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 Y axis, respectively, measured at p_i.

By varying the internal "force weights" in the active contour method these iterations converge to two different contours: 1) one covering both the palm and fingers and 2) one related to the palm only. In our original solution for the first (outer) contour the parameters were set to: $a = 0.8, b = 0.1, g = 0.6$. The second contour should detect the palm area of the hand, i.e. without covering the fingers. In this case we changed the parameter g to 0.15. Under these conditions the *elasticity* force is more dominant than in first case and this leads to a shorter contour than the first one. In order to lower the number of necessary iterations, compared to [3], we also changed the setting for a to $a = 0.1$ (Fig. 2). Thus the influence of the internal forces is getting lower and the contour approaches the image edges much faster.

The distances between contour points and the contour's mass center are computed next (Fig. 3). Thus we obtain two 1-D distance distributions. These distributions are subtracted one from the other, 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. 4).

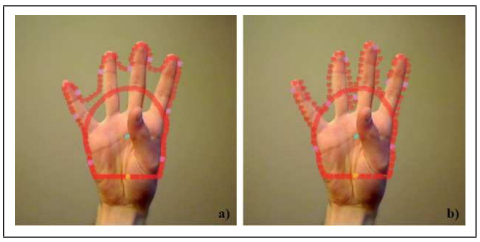

Fig. 2. Detecting two contours related to the hand: a) with a=0.8, b) with a=0.1.

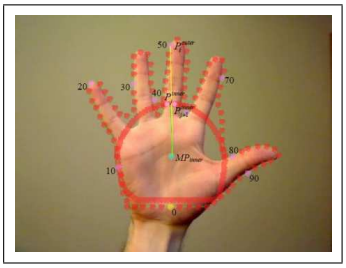

Fig. 3. Difference function generated for the two contours

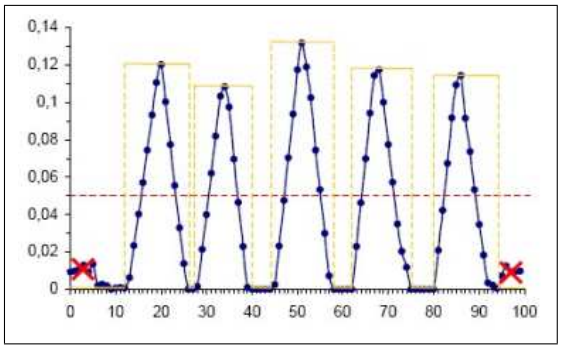

Fig. 4. Local maxima of the difference function

2.2 Hand poses and character set

The simple but quite reliable set of double contour-difference features allows for the detection of 21 static hand poses. One pose will be reserved for the *break* sign, that will terminate every gesture. To other poses we have assigned 20 characters of polish alphabet that appear most often in the language [8] (Fig. 5).

Diacritic characters (marked in yellow) are omitted, for first. By the way, the green letters does not appear in the polish finger alphabet PAP. Fig.

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%	Ż	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. 5. The expected frequencies of characters in polish language

6 illustrates the 20 characters assigned to static hand poses, i.e. these are characters in PAP: A, B, D or I or K or R, H or Y, L, W, and other most frequent characters.

The letter J has a similar meaning to I, but appears relatively seldom. Hence this letter will be represented by a "dynamic" version of the pose for letter I, similar to the one in PAP. The characters G and H are dynamic versions of W and Y, appropriately, as they are similar to signs in PAP. The termination sign corresponds to a "closed" fist pose.

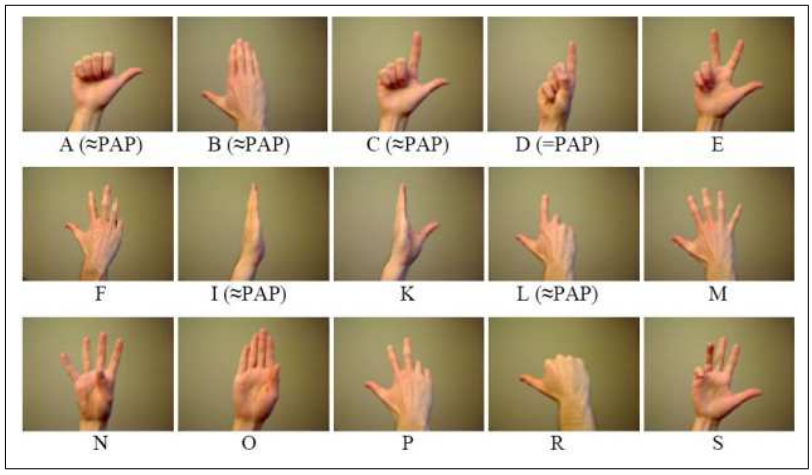

Fig. 6. Characters assigned to static poses

By detecting a hand motion in consecutive images we change the meaning of some characters, similarly to the PAP alphabet (Fig. 7). In this way diacritic characters are detected. (Ą, Ć, Ę, Ł, Ń, Ó, Ś, Ż). Additionally the remaining characters from the alphabet are detected, which have been omitted in the first single-image processing stage ("moving" I is J, W → G, Y → H) (Fig 8).

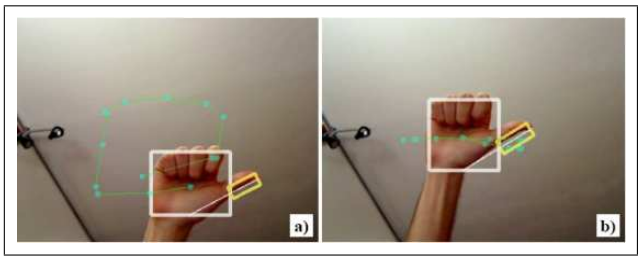

Fig. 7. Paths of the hand contour center detected in an image sequence: a) a random path, b) a letter Ą represented by a path and constant pose A.

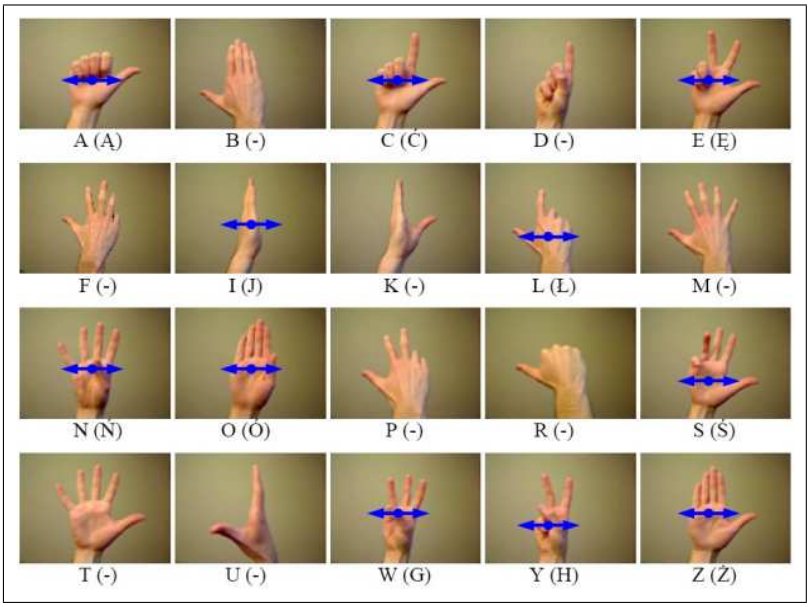

Fig. 8. Characters assigned to dynamic poses

3 Word model

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 [9]. Its special case, the left-to-right HMM, is useful to represent possible state paths that correspond to observation sequences [4] (Fig. 9).

The model is designed in two stages. First, the number of states and the model structure has to be decided. For every gesture a left-to-right sub-model is created (Fig. 10). The number of states matches the number of different

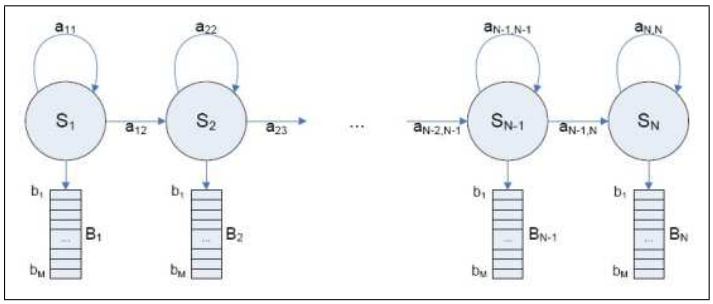

Fig. 9. The left-to-right Hidden Markov Model

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 [1], 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:

$$\sum_{o_i} \log \mathbf{P}(o_i|\lambda_j) \tag{3}$$

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.

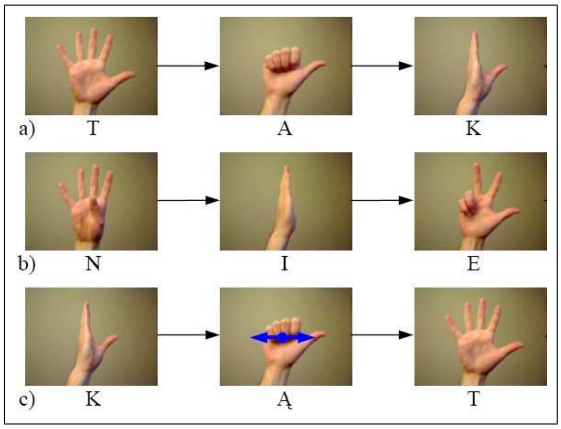

Fig. 10. Words as sequences of hand poses

The gol 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 current observation sequence:

$$\mathbf{P}(S^1S^2...S^tS^{t+1}|o^1o^2...o^to^{t+1}) =$$

$$= \mathbf{P}(o^{t+1}|S^{t+1})\ \mathbf{max}_{S^{t+1}}[\mathbf{P}(S^{t+1}|S^t)\mathbf{P}(S^1...S^t|o^1...o^t)] \quad (4)$$

4 Experiments

The system was implemented as a desktop application in Java language. The image sequences have been acquired by a low cost digital camera Logitech QuickCam Pro 9000, with a sensor matrix resolution of 1600 x 1200 pixel, automatic white balance, autofocus and automatic exposure time. To satisfy the on-line processing conditions a low-resolution image sequence has been acquired - frames of size 320 x 240 pixel, 24 bits RGB and 30 frames per second.

Some constraints have been posed ono the lighting conditions - although either natural or synthetic lighting has been applied, highlighting and dark conditions have been avoided. The scene has been 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.

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). In the testing stage other 160 sequences, acquired on-line, have been recognized. It appeared that the system works nearly perfectly if the poses are accurately shown, otherwise some misclassifications of single poses are possible.

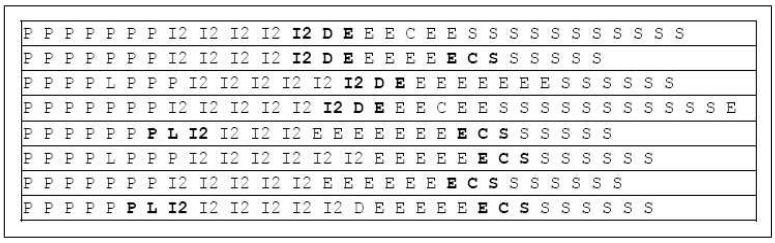

Fig. 11. Examples of measured sequences for the word "PIES"

The worst recognition rate of single characters (poses) was from around 75% (for such letters like K, T, O, D, *break*) to around 90% for the best recognized letters (e.g. N, I, E, P, S, A, C). To large amount these single letter errors have been compensated by the final gesture recognition stage as the word recognition rate reached nearly 95% (Fig. 11).

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 the range from 200 ms (letter I) to 400 ms (letter T). The gesture recognition stage with 5 words in our dictionary needs only few ms per frame.

5 Summary

In this paper our previous approach to hand sign recognition ([3]) has been extended to handle hand pose sequences with the speed of several frames/second on a typical PC. Two main stages, the hand tracking stage and the HMM-based gesture recognition, have been added. Some simplifications of the previously designed double active contour detection step allowed to speed up these expensive computations at the price of introducing larger pose classification errors. This drawback is now compensated with help of the context induced by the gesture model. Much effort was put reliably to recognize the "break" sign.

We also compared our static and dynamic pose categories, that we are able to recognize, with the hand poses defined in the PAP alphabet. The proposed approach is quite universal and different letters and words can be assigned, depending on application.

Acknowledgements

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

References

1. Baum L E, Petrie T, Soules G, Weiss N (1970), *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Statistics, 41:164–171, 1970, no. 1.
2. Kapuściski T (2006), *Rozpoznawanie polskiego języka miganego w systemie wizyjnym*, Ph.D. Thesis, Wydział Elektrotechniki, Informatyki i Telekomunikacji, Uniwersytet Zielonogórski, Zielona Góra, 2006, http://zbc.uz.zgora.pl/Content/3896/book.pdf.
3. Kasprzak W, Skrzyński P (2006), *Hand image interpretation based on double active contour tracking*, In: Zielińska T, Zieliński C (eds) *ROMANSY 16. Robot design, dynamics, and control*, CISM Courses and lectures - No. 487, Springer, Wien NewYork, 439-446.
4. Kasprzak W (2009), *Rozpoznawanie Obrazów i Sygnałów Mowy*, WUT press, Warszawa, 2009.
5. Kass M, Witkin A, Terzopoulos D (1998), *Snakes Active contour models*, International Journal of Computer Vision, 1:321–331, 1998, no. 4.

6. Marnik J (2003) *Rozpoznawanie znaków Polskiego Alfabetu Palcowego*, StatSoft Polska, 2003, http://www.statsoft.pl/czytelnia/badanianaukowe/d0ogol/marnik.pdf.
7. Murphy K P (2002), *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Thesis, UC Berkeley, 2002.
8. Przepiórkowski A (2006), *Frekwencja liter w polskich tekstach*, Poradnia językowa PWN, 2006, http://poradnia.pwn.pl/lista.php?id=7072.
9. Rabiner L, Juang B (1993), *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
10. Wilkowski A (2008), *An Efficient System for Continuous Hand Posture Recognition in Video Sequences*, In: Rutkowski L, Tadeusiewicz R, Zadeh L A, Zurada J,*Computational Intelligence: Methods and Applications*, 411–422, EXIT, Warszawa.
11. Xu C-Y, Prince J L (1998), *Snakes, Shapes, and Gradient Vector Flow*, IEEE Transactions on Image Processing, 7:359–369, 1998, no. 3.