ROMANSY 16. Robot Design, Dynamics, And Control.
CISM Courses and Lectures, No. 487.

Hand image interpretation based on double active contour tracking

Włodzimierz Kasprzak [†][*] and Piotr Skrzynski [†]
[†] Warsaw University of Technology, Institute of Control and Computation Eng.,
ul. Nowowiejska 15/19, PL-00-665 Warsaw, Poland
e-mail: W.Kasprzak@ia.pw.edu.pl

Abstract. We propose an approach to hand sign interpretation in image that is based on active contour tracking. We can decompose our approach into 5 steps: a color-based skin pixel detection, a double hand contour detection, the localization of fingers and palm (the hand description generation), the detection of a final position (with respect to considered signs) and finally, the interpretation of a single position or a sequence of positions in terms of a hand sign. We employ a double active contour-based finger and palm localization in the image and a subsequent interpretation in terms of signs. As a final result, 21 different signs are recognized, that correspond to hand positions (i.e. the visibility of palm, fingers and thumb).

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). In this paper we are interested in man-machine communication aspects, assuming that a single view is available, the hand can be freely oriented in space, while finger positions can also be different. Our goal is to interpret the recognized hand in terms of commands given to the machine by a human operator. The creation of such image recognition system requires solutions in several image analysis areas: color-based region-of-interest detection (Yining et al., 1999; Fu et al., 2000), hand contour detection, 3-D hand instance generation and interpretation.

A straightforward approach to object detection is to extract its shape (or contour) information from the image. A popular approach to closed contour detection for free-form and deformable objects uses "active contours" (Kass et al., 1988) or "snakes" (Berger, 1990). Snakes (active contours) (Kass et al., 1988; Berger, 1990; Osher and Paragios, 2003) are curves defined in image plane that can change their shape and can move under the influence of forces. Typically we consider parametric curves and allow them to move toward desired features, usually edges, under the influence of *external* and *internal* forces.

[*]The authors gratefully acknowledge the support from the Faculty for Electronics and Information Technology, Warsaw University of Technology.

There are two key problems with parametric active contours. First, if the initial contour is not sufficiently close to the true boundary it will likely converge to a wrong result. The basic idea is to increase the capture range of the external force fields and to guide the contour toward the desired boundary (Cohen and Cohen, 1993). The second problem is that active contours only partially progress into boundary concavities (Abrantes and Marques, 1996). In our work we adapt the approach of GVF snakes (Xu and Prince, 1998), in which external forces for active contour models take the form of gradient vector flow (GVF) fields. Particular advantage of the GVF snake over a traditional snake is its ability to move into boundary concavities.

The final contour should be interpreted in terms of a 3-D object. In this paper we compute two different active contours and analyze their differences in terms of finger and palm detection, i.e. we analyze the waveform of a 1-D difference function, that expresses contour point distances from the contour's center of mass (Kauppinen et al., 1995).

2 Color-based ROI detection

The sensor data is 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, like YUV, HSV and YCbCr, we found the YCbCr space to be most suitable for our task. In this scheme the Y component represents intensity, the Cb component - the "chromatic distance" of given color from pure blue color, and the Cr component - such a distance from pure red color.

We applied a normalization of the Cb and Cr components driven by the intensity Y normalization to a predefined value (let us fix it to 128). Due to this procedure we can achieve very stable and narrow intervals around 100 (for Cb) and 150 (for Cr) that represent the skin color.

$$\begin{bmatrix} Y_{norm} \\ Cb_{norm} \\ Cr_{norm} \end{bmatrix} = \begin{bmatrix} 128 \\ (Cb - 128) * 128/Y + 128 \\ (Cr - 128) * 128/Y + 128 \end{bmatrix} \tag{2.1}$$

Without the intensity-based normalization a 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 case of a very structured background containing colors similar to the skin color, detection problems may appear. Obviously other color correction solutions may be tried, like for example density or luminance regularization (Chai and Nagan, 1999).

3 Active contour detection

Snakes (active contours) (Kass et al., 1988; Berger, 1990; Osher and Paragios, 2003) are curves defined in image plane that can change their shape and can move under the influence of forces. These are decomposed into *internal* forces, as a result of expected curve's *stiffness*, and *external* forces computed from the image data and distributed over the image. After the snake is initialized inside-of 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 a large bending (*bending* 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 $p_i = (x_i, y_i)$, $(i = 0, ..., n-1)$ and connecting line segments $L_i = [p_i, p_{i+1}]$. During contour evolution the total energy of points is minimized.

3.1 The GVF snake

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

$$E_{\text{ext}} = K_0 * \sum_{i=0}^{n-1} I(x_i, y_i), \tag{3.1}$$

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.

We have applied another approach for the computation of external forces, called *gradient vector field* (GVF) (Xu and Prince, 1998), that allows us better than before to move 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 (Figure 1). The energy functional is similar to the one formulated for the computation of optical flow (Horn and Schunck, 1981):

$$\mathcal{E} = \int \int m(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 \cdot |\mathbf{v} - \nabla f|^2 dxdy \tag{3.2}$$

where $f(x, y) = \nabla I(x, y)$ is the gradient image and m is some weight parameter.

The minimization of $\mathcal{E}$ is achieved by solving a pair of decoupled linear partial differential equations that diffuses the gradient vectors of a gray-level or binary edge map computed from the image.

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

$$E_{\text{int}} = E_{\text{elastic}} + E_{\text{stiffness}} \tag{3.3}$$

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

$$E_{\text{elastic}} = K_1 * \sum_{i=0}^{n-1} |p_i - p_{i-1}|^2, \tag{3.4}$$

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 in control points:

$$E_{\text{stiffness}} = K_2 * \sum_{i=0}^{n-1} |p_{i-1} - 2p_i + p_{i+1}|^2, \tag{3.5}$$

where K_2 is an 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 the balance of forces for given contour in given image is achieved:

$$\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}} = 0 \tag{3.6}$$

We repeatedly apply for every control point p_i the pair of equations:

$$x_i = x_i + a \cdot F_{\text{elastic}}(X, i) + b \cdot F_{\text{stiff}}(X, i) + g \cdot F_{\text{ext}}(X, i) \tag{3.7}$$

$$y_i = y_i + a \cdot F_{\text{elastic}}(Y, i) + b \cdot F_{\text{stiff}}(Y, i) + g \cdot F_{\text{ext}}(Y, i) \tag{3.8}$$

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, at control point p_i.

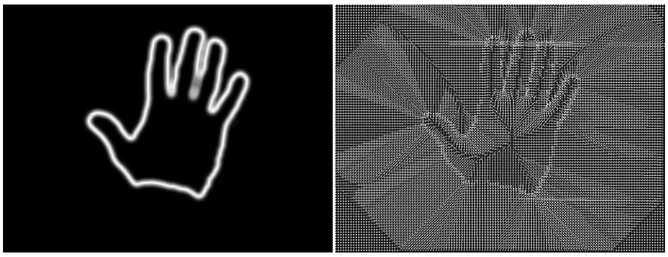


Figure 1. The gradient image (on the left) and the gradient vector field (with $m = 0.2$) (on the right) computed from a hand image with a uniform background.

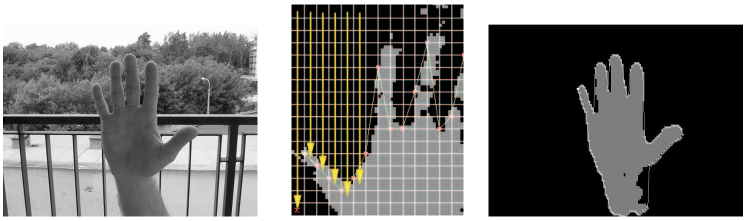
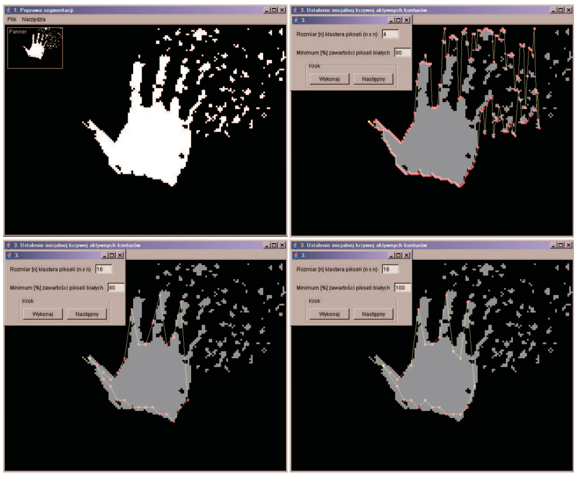


Figure 2. The detection of initial active contour in the segmented image: (a) sample image, (b) the detection of consecutive initial contour points, (c) examples of a well-detected initial contour.

3.2 Initialization of the active contour

The input data to this step is the binary image containing skin-color pixels. The binary image is divided into 3x3 or 4x4 blocks (the block size is a parameter). At first the blocks are examined from left to right and from top to down. If a block contains a sufficient amount of skin pixels (let us call it the "block filling" threshold) then the next


1. Poprawa segmentacji
Plik
Narzędzia
Panner
3. Ustalenie inicjalnej krzywej aktywnych konturów
3.
Krok
Wykonaj
Następny
3. Ustalenie inicjalnej krzywej aktywnych konturów
3.
Krok
Wykonaj
Następny
3. Ustalenie inicjalnej krzywej aktywnych konturów
3.
Krok
Wykonaj
Następny

Rozmiar [n] klastera pikseli (n x n) 4
Minimum [%] zawartości pikseli białych 80

Rozmiar [n] klastera pikseli (n x n) 16
Minimum [%] zawartości pikseli białych 80

Rozmiar [n] klastera pikseli (n x n) 16
Minimum [%] zawartości pikseli białych 100

Figure 3. The detection of initial active contour in a deteriorated segmentation image - its dependence from the block size and the "block filling" threshold.

active contour point is located in the left upper corner of this block. The search moves to the next column on the right side. This cycle is iterated until the end of image or a no-skin column is found (see Figure 2). Next, the blocks are examined from right to left and from bottom to up. An eventual control point of the active contour is now located in the bottom right corner of the examined block.

There is no guarantee that the initial contour matches sufficiently exact the hand shape. In a deteriorated segmentation image (Figure 3) different block sizes and "block filling" parameters may lead to correct or quite incorrect initial contours.

3.3 Outer (i.e. hand) and inner (i.e. palm) contour detection

Two active contours are computed out from the same initialization. In the image of size 640 × 480 in our experiments the hand covered the image part of 7 - 20% . For the first (outer hand) contour the *snake* 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 initial one (Figure 4).

4 Contour difference description

At first a new relocation of contour's control points is established due to an approximation of existing points. The new control points are uniformly distributed and its number is fixed. Next the center of mass of the inner contour is determined. This center point together with the direction of the longer boundary box of the outer contour allows for a first contour post-processing step - the elimination of a forearm part (Figure 5).

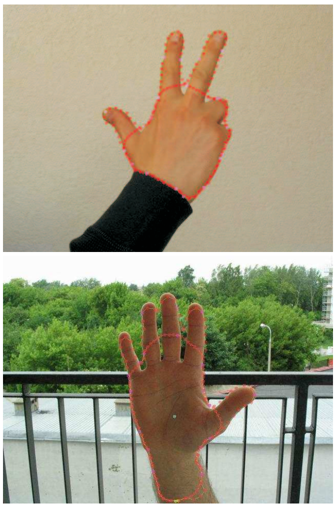


Figure 4. Typical results of double active contour detection.

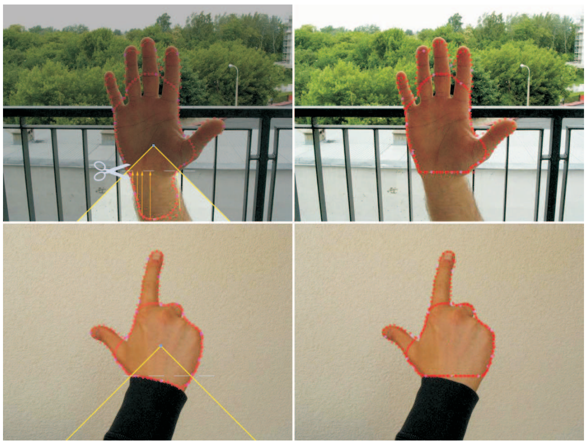


Figure 5. The elimination of a forearm part from both contours: (left column) the part contained in the triangle area is eliminated, (right column) the post-processed contours.

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. 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 (Figure 6).

5 Hand instance detection and sign interpretation

Finally a hand instance detection on base of two contour descriptions can be performed and this hand instance can then be interpreted in terms of a specific sign from a set of 21 static signs. The implicit hand model consists of a box representing the palm and 5 cylinders that represent the 5 fingers.

Current hand instance is generated out 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 (Figure 7(a)). The fingers are detected on base of 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 (Figure 7(b)). The thumb is detected among the points if its tip projection onto the palm rectangle is located below the mass center point (Figure 7(c)).

The relation of inner contour's height-to-width induces the front or side-view, while the height-to-height relation for both contours induces the fraction of top-view.

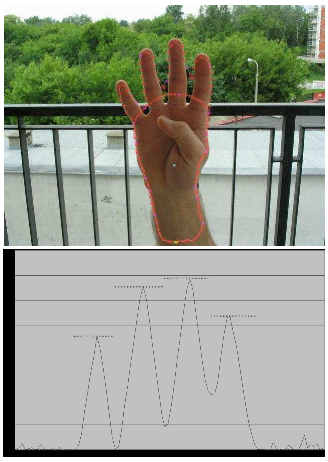
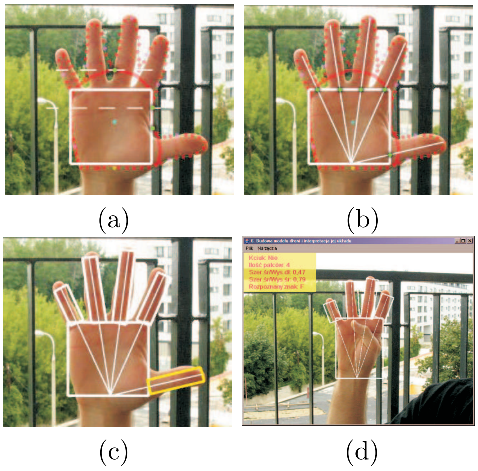


Figure 6. Example of the difference between distance-to-mass center values for both active contours. We have here 5 easy-detectable extremum points along the horizontal axis.



(a) (b)

(c) (d)

Figure 7. The detected hand instance and its corresponding sign: (a) palm detection, (b) finger detection, (c) thumb detection, (d) the GUI of the application showing the final sign and hand description.

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

Sign	Thumb position	Detected fingers	Hand view
A	No	closed (0)	Front/Back
B	No	5in1 (0)	Front/Back
C	No	1	Front/Back
D	No	2	Front/Back
E	No	3	Front/Back
F	No	4	Front/Back
G	No	5in1 (0, 1)	Side
H	Left	thumb (1)	Front/Back
I	Left	4in1 + thumb	Front/Back
J	Left	2	Front/Back
K	Left	3	Front/Back
L	Left	4	Front/Back
M	Left	5	Front/Back
N	Left	4in1 + thumb	Side
O	Right	thumb (1)	Front/Back
P	Right	4in1 + thumb	Front/Back
Q	Right	2	Front/Back
R	Right	3	Front/Back
S	Right	4	Front/Back
T	Right	5	Front/Back
U	Right	4in1 + thumb	Side

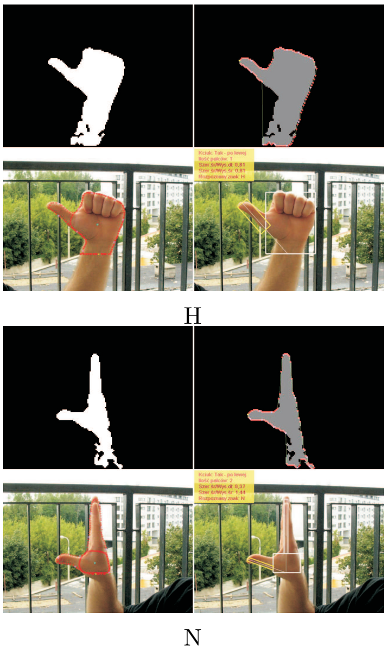


Figure 8. Some steps of sign detection of letters H and N.

6 Test examples

We have experimentally verified that our hand description can in an unique way be transformed into a set of 21 signs (Table 1). In Figure 8 examples of intermediate steps of our hand sign detection approach are shown. If the fingers are well separated one from the other then the first active contour is progressing in-between them. The second active contour is then matching the boundary palm area of the hand. Otherwise, both contours are similar, except of the visible thumb position.

Bibliography

J. M. Rehg, and T. Kanade. Digit Eyes: Vision-Based Human Hand Tracking. *Report CMU-CS-93-220*, School of Computer Science, Carnegie Mellon University, 1993.

D. Yining, B. S Manjunath, and H. Shin. Colour image segmentation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 2:2446–2451, 1999.

C.-S. Fu, W. Cho, and S. K. Essig. Hierarchical colour image region segmentation for content-based image retrieval system. *IEEE Transactions on Image Processing*, 9:156–162, 2000, No.1.

M. Kass, A. Witkin, and Terzopoulos D.. Snakes: Active Contour Models. *International Journal of Computer Vision*, 18:321–331, 1988, No. 1.

M. O. Berger. Snake growing. *Lecture Notes in ComputerScience*, 427:387–396, 1990.

S. Osher, and N. Paragios, Editors. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer Verlag, New York, 2003.

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

R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos. Biometric Identification through Hand Geometry Measurements, *Biometrics*, 22, 2000, No. 10.

C-Y. Xu, and J. L. Prince. Snakes, Shapes and Gradient Vector Flow. *IEEE Trans. on Image Processing*, 7:359–369, 1998, No. 3.

L. D. Cohen, and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:1131–1147, 1993, No. 11.

A. J. Abrantes, and J. S. Marques. A class of constrained clustering algorithms for object boundary extraction. *IEEE Trans. on Image Processing*, 5:1507–1521, 1996, No. 11.

D. Chai, and K. N. Nagan. Face Segmentation Using Skin Color Map In Videophone Applications. *IEEE Trans. on Circuits and Systems for Video Technology*, 1999. (www.soem.ecu.edu.au/ dchai/public/papers/ csvt99.ps).

B. K. P. Horn, and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.