Vision-Based Algorithm for Real-Time Hand Posture Recognition

Bojan MARKOVICH
Faculty of Mathematics
University of Belgrade
Studentski trg 16, Belgrade
SERBIA
markovich.bojan@gmail.com

Milan TUBA
Faculty of Computer Science
Megatrend University Belgrade
Bulevar umetnosti 29, N. Belgrade
SERBIA
tuba@ieee.org

Abstract: - Importance of human-computer interaction (HCI) systems is increasing since they revolutionized the way people interact with computers. Gesture and posture analysis, especially hand gesture and posture recognition, is one of the more recent areas in the HCI with ever widening field of applications. In this paper we propose an algorithm for hand posture recognition that is based on skin detection and Hu moments. Our algorithm is invariant to the hand position and experiments proved it to be robust with high percentage of posture recognition. Algorithm works in real-time and can successfully be used in variety of applications.

Key-words: - Hand posture recognition, Hand segmentation, Posture-based HCI, Image processing

1 Introduction
Computing is no longer constrained to the use of desktop computers and notebooks. Some new application environments include virtual and augmented reality, interaction with large displays, enhanced visualization of large data collections. That is one of the reasons why there has been growing interest in the development of new approaches and technologies for bridging the human–computer barrier.

Posture and gesture recognition can be seen as a way for computers to begin to understand human body language. Among different body parts, the hand is the most effective, general-purpose interaction tool due to its proficient functionality in communication and manipulation, thus it is normal that the most powerful and natural interface for human-computer interaction is the hand posture and gesture. Also, there are some promising virtual environment applications such as surgical simulations and training systems that incorporate hand postures and gestures [1].

To interpret postures, computers need to perceive the outside world. The use of video cameras for tracking hand movements is the most natural way, but also the most challenging. In that way, computer vision represents a promising alternative to data gloves because of its potential to provide more natural, contactless interaction. In computer vision, hand posture recognition generally involves various stages like video acquisition, background subtraction, feature extraction and posture recognition.

We propose here an algorithm that is designed for real-time hand posture recognition. It implements a skin-based hand segmentation and conducts shape analysis on the segmented area.

The rest of the paper is organized as follows. In Section 2 we present related work and Section 3 is an overview of shape analysis, specifically about contours and it also introduces terms later used. Next, in Section 4 we describe our proposed algorithm and in Section 5 we discuss achieved results. Section 6 concludes the paper.

2 Related work
Image processing is very active research area with many different applications [2], [3]. In recent years a lot of work have been done in the field of posture and gesture recognition for human computer interface (HCI). Various techniques and algorithms were proposed and different results were achieved. Since the results are not fully acceptable for real life usage, the field which is dealing with this problem is still very active and in progress.

In [4] authors are considering a vision-based system that can interpret a user’s gestures in real time to manipulate windows and objects within a graphical user interface. First, with hand segmentation procedure binary hand blob is extracted and Fourier descriptors are used to represent the shape of extracted blobs. Authors then used radial-basis function network for pose classification whose output is used along with

This research is supported by Ministry of Science, Republic of Serbia, Project No. 44006
motion information as input for gesture recognition using hidden Markov models and/or recurrent neural networks. Gesture recognition results are around 90% and real time processing rates up to 22 frames per second were obtained.

Yin and Xie have presented a posture recognition system for the purpose of controlling a real humanoid robot [5]. System applies RCE neural network based color segmentation algorithm to separate hand images from complex background and then the topological features of the hand are extracted from the silhouette of the segmented hand region.

Vision based recognition system based on hand gesture can be used for personal authentication where hand gesture is used as a password [6].

In [7] authors propose a real time vision system for hand gesture based computer interaction to control an event like navigation of slides in Power-Point presentation. As opposed to previously discussed studies, Gaussian Mixture Model (GMM) was used to extract hand from the video sequence. Extreme points were extracted from the segmented hand using star skeletonization and recognition was performed by distance signature.

Some new visual representations for hand motion are proposed like one in [8] based on the motion divergence fields. A large collection of gesture video samples is required but proposed method achieves higher recognition accuracy then other state-of-the-art motion and spatio-temporal features with average recognition time of 35 msec per gesture.

A novel approach for hand gesture recognition based on Gabor filters and SVM classifiers for environments with varying illumination is presented [9]. Suggested approach is robust against varying illumination, which is achieved using an adaptive skin-color model switching method. It is insensitive to hand-pose variations, which is achieved using a Gabor filter based gesture angle estimation and correction method and it also allows users to wear either a long or short-sleeve shirt, which is achieved using a method that segments the hand from the forearm. It was tested in realistic conditions and recognition rate of 96.1% was achieved.

There are systems able to localize in real time a hand with variable finger configurations in images with complex backgrounds, different lighting conditions and different positions of the hand with respect to the camera. Cluttered backgrounds and thick textured images which usually make it hard to compare edge information with silhouette models are dealt by simultaneously using connected curves and topological information [10].

There are hand gesture recognition systems based on local linear embedding [11] as well as ones involving changing shapes and trajectories, using a Predictive Eigen Tracker [12]. Some interesting attempts are made in the field of hand gesture recognition using neural networks and hidden Markov models in [13], [14].

Just and Marcel in [15] address the problem of nonavailability of standard public hand gesture databases. The result is that algorithms proposed nowadays for hand gesture recognition are not evaluated on common data. They compare two state-of-the-art algorithms for hand gesture recognition HMM and IOHMM on the same database.

3 Shape analysis
Shape can be characterized by its area, centroid, contour, convex hull, convexity defects, etc. Its determination usually involves segmentation and edge analysis [16], [17]. A contour is a list of pixels that represent a curve on an image. Edge detection filters can be used to find the edge pixels that separate different segments in an image but they don’t give any information about those edges as entities. What we need is to be able to assemble those edge pixels into contours in order to extract various contour features and gain some knowledge about them.

One of the contour retrieval methods have been presented by Suzuki and Abe in [18]. This border following algorithm extracts contours and determines the surroundness relations among the borders of a binary image constructing an contour tree along the way.

While working with contours there are many different things we might want to do with them. Common tasks include characterizing the contours in various ways, simplifying or approximating them, matching them to templates and with each other, etc.

When we conduct shape analysis, we usually approximate a contour representing a polygon with another contour having fewer vertices. One way to do that is popular Douglas-Peucker (DP) approximation where the goal is to find a similar curve with fewer points given a curve composed of line segments. The algorithm defines dissimilarity measure based on the maximum distance between the original curve and the simplified curve and at the end the simplified curve consists of a subset of the points that defined the original curve. We are able to control
the number of points on simplified curve changing mentioned maximum allowed distance. This algorithm is also known as iterative end-point fit algorithm or the split-and-merge algorithm. Many of algorithms for approximation used nowadays are improvements of this algorithm.

The most common task associated with contours is matching them in some way with one another. One of the ways to compare two contours is to compute their characteristic called contour moments computed by integrating over all of the pixels of the contour.

Moment \((p, q)\) of a contour is defined as

\[
m_{p,q} = \sum_{i=1}^{n} I(x, y)x^p y^q
\]

where \(p\) is the order of \(x\) and \(q\) is the order of \(y\).

A central moment is basically the same as the moments just described except that the values of \(x\) and \(y\) used in the formulas are displaced by the mean values

\[
\mu_{p,q} = \sum_{i=1}^{n} I(x, y)(x-x_{\text{avg}})^p (y-y_{\text{avg}})^q
\]

where \(x_{\text{avg}} = \frac{m_{10}}{m_{00}}\) and \(y_{\text{avg}} = \frac{m_{01}}{m_{00}}\).

The normalized moments are the same as the central moments except that they are all divided by an appropriate power of \(m_{00}\)

\[
\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{00}^{p+q}}
\]

Hu invariant moments are very important since they are scale and rotation invariant contour descriptors.

### 4 Our Hand Recognition Algorithm

Our algorithm is designed for real-time usage. It implements a skin-based hand segmentation and conducts shape analysis on the segmented area, or more precisely, shape’s contour feature and it’s Hu invariant moments as contour descriptors. After that, it classifies shown posture using developed model and additional stabilization is performed. Algorithm has several phases which can be seen on Figure 1.

![Algorithm phases diagram](image)

**Video capture**: Video is being captured from web camera frame by frame which brings us into the image processing field. Since we are doing skin-based hand segmentation in the next step transforming image color space into YCrCb is done. It has been shown that RGB color space is not so good for recognizing skin color and that for that task YCrCb color space is more adequate.

**Hand recognition**: It is empirically found that skin color in YCrCb color space lies between following boundaries

\[
50 \leq Y \leq 255
\]

\[
130 \leq Cr \leq 185
\]

\[
80 \leq Cb \leq 135
\]

Every pixel that satisfies upper conditions is set to white color and every other to black, having in result binary picture with marked hand region.
Because we are working with real time images with a lot of noise, we apply erode and dilate filter. The Erode filter, commonly known as shrink or reduce, removes islands of pixels smaller than the structural element (kernel) in a binary or grayscale image while dilate filter, commonly known as fill, expand, or grow, fills holes smaller than the structural element (kernel). The results produced are smooth contours, breaking of narrow isthmuses, and elimination of small islands and sharp peaks or capes in an image. Now, the image is ready for feature extraction.

**Feature extraction:** We are characterizing shape of the hand by finding it’s contour. Contour can have too many points and that can be bad for generalization of contour. Later on we are calculating Hu invariant moments and that procedure is taking all of the contour points into account so we are considering speed too. That is the reason why the next step is approximating obtained contour with a polygon with less number of points.

**Posture classification:** Hand can be on various distances from camera and in different angles. Hu moments are invariant to scale and rotation and represent a good choice as contour descriptors in our case. Hu invariant moments are calculated (Eq. 1) and the next step is comparing contour with all contours in our prepared representation model of different hand postures. After some testing chosen metric is

\[
C(X,Y) = \sum_{i=1}^{7} \left| \frac{m_i^X}{m_i^Y} - \frac{1}{m_i^Y} \right| \tag{2}
\]

where \(m_i^X\) and \(m_i^Y\) are

\[
m_i^X = \text{sign}(h_i^X) \log|h_i^X| \quad \text{and} \quad m_i^Y = \text{sign}(h_i^Y) \log|h_i^Y|
\]

and \(h_i^X\) and \(h_i^Y\) are the Hu moments of X and Y contour respectively.

The smallest distance result is found and current contour is classified as appropriate hand posture.

**State machine:** We are not taking movement of the hand into account thus we are only recognizing static hand gestures which can be seen as hand postures. Still, it is preferred that the state of a recognized hand gesture is not changing on frame bases and to be more stable. We have implemented buffer of last 5 frame recognitions and state of the hand recognition is changing based on majority of different hand recognitions. This approach causes a lag of time for 3 frames, between the moment when new hand gesture is shown and the moment when the state is changed. But more valuable side is that recognition is more stable and it gives us good base for later recognition of dynamic hand gestures.

We have 7 hand postures saved in our recognition model of different hand postures. The model is formed averaging 10 different images of the same posture per posture. Number of points on this posture model contours after approximation can be seen in Table 1.

<table>
<thead>
<tr>
<th>Hand posture</th>
<th>No. of points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand</td>
<td>36</td>
</tr>
<tr>
<td>Three</td>
<td>36</td>
</tr>
<tr>
<td>Victory</td>
<td>33</td>
</tr>
<tr>
<td>Point</td>
<td>38</td>
</tr>
<tr>
<td>Fist</td>
<td>37</td>
</tr>
<tr>
<td>Pick</td>
<td>36</td>
</tr>
<tr>
<td>Horns</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 1. Number of points on different posture contours after polygon approximation

Also, in Table 2, we can see calculated Hu moments for this posture models. Besides the fact that the moments from dissimilar postures are pretty different, we can see the order of each Hu moment and better understand why we used formula (2) and logarithmic function in matching contours.

The software that has been developed which allows real time posture recognition and provides implementation of the described algorithm.

<table>
<thead>
<tr>
<th>Posture model</th>
<th>HU invariant moments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h1</td>
</tr>
<tr>
<td>Hand</td>
<td>1.983E-1</td>
</tr>
<tr>
<td>Three</td>
<td>2.13E-1</td>
</tr>
<tr>
<td>Victory</td>
<td>2.426E-1</td>
</tr>
<tr>
<td>Point</td>
<td>2.141E-1</td>
</tr>
<tr>
<td>Fist</td>
<td>1.659E-1</td>
</tr>
<tr>
<td>Pick</td>
<td>2.284E-1</td>
</tr>
<tr>
<td>Horns</td>
<td>2.351E-1</td>
</tr>
</tbody>
</table>

Table 2. Hu moments of different posture models
5 Results

Proposed algorithm was tested under controlled lighting conditions and on plain non-moving background. Since no wrist cutting method is implemented, only long sleeves were allowed. For best results it is necessary that the hand’s plane is parallel to camera’s plane. We had paid attention to avoid hand self-occlusions. Occlusions are still one of the biggest problems in almost every vision-based system.

On Picture 1, we can see all possible hand postures after hand recognition step.

Picture 1. Hand postures after hand recognition step: (a) hand, (b) point, (c) three, (d) victory, (e) horns, (f) fist, (g) pick

Each hand posture was tested 30 times on the different distances from the camera and in the various angles. Achieved results are presented in Table 3.

<table>
<thead>
<tr>
<th>Posture</th>
<th>Correct</th>
<th>Wrong</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand</td>
<td>23</td>
<td>7</td>
<td>77%</td>
</tr>
<tr>
<td>Three</td>
<td>22</td>
<td>8</td>
<td>73%</td>
</tr>
<tr>
<td>Victory</td>
<td>27</td>
<td>3</td>
<td>90%</td>
</tr>
<tr>
<td>Point</td>
<td>24</td>
<td>6</td>
<td>80%</td>
</tr>
<tr>
<td>Fist</td>
<td>29</td>
<td>1</td>
<td>97%</td>
</tr>
<tr>
<td>Pick</td>
<td>17</td>
<td>13</td>
<td>57%</td>
</tr>
<tr>
<td>Horns</td>
<td>21</td>
<td>9</td>
<td>70%</td>
</tr>
</tbody>
</table>

Table 3. Hand recognition results

Overall achieved recognition rate is around 78%. Table 3, among other information tells us that the best classification are having shapes with the most viewpoint invariant features, fist and victory posture. Also, the worst result has pick posture where there is plenty opportunity for self-occlusion.

Our software in action with testing phase GUI can be seen on Picture 2.

<table>
<thead>
<tr>
<th>Posture model</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand</td>
<td>0.160</td>
</tr>
<tr>
<td>Three</td>
<td>0.230</td>
</tr>
<tr>
<td>Victory</td>
<td>0.475</td>
</tr>
<tr>
<td>Point</td>
<td>0.371</td>
</tr>
<tr>
<td>Fist</td>
<td>0.276</td>
</tr>
<tr>
<td>Pick</td>
<td>0.367</td>
</tr>
<tr>
<td>Horns</td>
<td>0.384</td>
</tr>
</tbody>
</table>

Table 4. Distance from model related to Picture 2.

We can see that current shown posture distance from posture models is the smallest from hand posture model and as a result we have correctly classified posture as “Hand”. The second closest one is three posture model and we can observe that the achieved results of these two postures in Table 3. have very close success percentage.

Shown posture distance from victory posture model is the smallest and posture is correctly classified.

Even a fist recognition rate is 97%, when long sleeves are not used (Picture 3.) we have false classification. This shows importance of wrist cutting method.

Picture 2. Real time “hand posture” recognition

Picture 3. Short-sleeves hand posture recognition

6 Conclusion

Hand posture detection is very important for contemporary HCI systems. We proposed and developed an efficient and robust algorithm for real-time hand posture detection. It segments binarized image and detects hand and subsequently recognizes
hand posture by using Hu moments. This position invariant method showed in our experiments significant success rate. It was rather insensitive to light changes, movements and hand position.

Further development will include posture and gesture recognition for more versatile system and improvement in detection rate by precise calibration of the existing software system.

References