Geometry-based Dynamic Hand Gesture Recognition

Duc-Hoang Vo, Huu-Hung Huynh, and Jean Meunier

Abstract—Hand gestures play an important role in communication in the hard-of-hearing community. They are used to convey information instead of words. Besides, a system which is developed to identify gestures can be also used for human-computer interaction. In this paper, we propose a vision-based approach for recognizing dynamic hand gestures. Our processing consists of three main stages: pre-processing, feature extraction and recognition. The first stage involves two sub-stages: segmentation which locates the hand and extracts the corresponding silhouette using color information; separation that removes the arm based on geometrical properties. Some characteristics which describe the hand posture are then extracted. Finally, the recognition is performed using two popular algorithms, which are Dynamic Time Warping and Hidden Markov Model. The experiment is conducted on SKIG dataset with a comparison of classification accuracies corresponding to the two mentioned methods.

Index Terms—gesture, fingertip, wrist, cross section, skin filter, codeword, geometrical property.

1. Introduction

In the research field of computer vision, hand gesture recognition is a well-known topic with a wide range of applications. Systems used as supported devices, which help the deaf people in communication, are also based on this common platform. The problem of recognizing hand gesture is studied in two trends, corresponding to two data types that are static and dynamic gestures. The static gesture is usually described via the information of hand shapes and poses, while the hand motion is determined to represent the dynamic type. However, there is a close relationship between these trends, since many approaches of dynamic gesture identification are developed based on static gesture recognition techniques.

In this paper, a vision-based approach is proposed for dynamic hand gesture classification in which the features are extracted based on the geometrical properties of the hand silhouette, the Dynamic Time Warping (DTW) and Hidden Markov Model (HMM) algorithms are used for recognition in order to give a comparison of accuracies. The rest of the paper is organized as follows: Section 2 describes some related work; the details of the proposed approach are presented in Section 3; the experimental results are shown in Section 4 and Section 5 gives the conclusion and discussion.

2. Related Work

Many researches on hand gesture recognition have been implemented recently. For example, the approach in [1] was proposed for identifying sign language using a multi-stream HMM technique combined with weighted information of hand position and movement. Another system was developed using skin color segmentation and artificial neural network [2]. However, the extracted features described in these methods do not contain much information of the shape of hand silhouette, an important attribute which affects recognition accuracy, thus the level of generalization is not really high.

Another type of hand gesture applications is human-computer interaction. For example, a device was developed to replace the computer mouse for the disabled [3]. The authors in [4] presented a system which allows user to control TV and DVD player via a camera. In [5], an algorithm which automatically identifies a limited set of hand gestures was proposed to control robot performing tasks. The high computational cost is the general limitation of these methods. In addition, the solution of [6] can support user interaction with multimedia systems using a glove. However, it is inconvenient for the user to wear an expensive glove.

In the recognition problem, extracting feature greatly affects the recognition accuracy. Most features used in related researches were extracted from the three following methods.

2.1. Model-based

This approach uses some kinematic parameters and projecting its edges onto a 2D space to create a 3D hand model. A search in the parameter space for the best match between projected edges and the edges acquired from the input image is performed to estimate kinematic values. In [7], a set of multi-viewpoint images was used for controlling objects in virtual world. The information extracted from the hand shape and movement was used to recognize eight kinds of commands. In another work,
the authors used joint angle values for manipulating in virtual space [8]. The hand region was determined from multiple images which acquired via a multi-viewpoint camera system. These silhouette images were integrated to reconstruct the hand pose, and then all joint angles were estimated. In model-based approaches, initializing parameters which are close to the solution at each frame is a challenge, and such systems require more time for designing.

2.2. View-based

In this approach, the hand is modeled based on a collection of intensity images. At each time, gestures are represented as a sequence of views. The principal component analysis technique is often used to provide an efficient representation of a large set of high-dimensional points using a small set of orthogonal basis vectors [9]. However, a large amount of data is required for training stage. Besides, the cost for building such systems is also an issue.

2.3. Low-Level Features

Some studies proposed a relative feature space, in which the detailed information of hand shape is not necessary. The used characteristics consist of hand position (centroid coordinates), the angle of axis of least inertia, and eccentricity of the bounding ellipse [5]. Such approach only obtains high efficiency when applied on gestures, which are very different in visual appearance. In this paper, this feature type is used to represent hand posture information.

3. Proposed Approach

In this section, the three main stages of our methodology, namely pre-processing, feature extraction, and recognition, are presented. The flowchart of our approach is given in Fig. 1.

3.1. Pre-processing

There are necessary steps to segment the hand from the original frame.

3.1.1. Skin segmentation

Two common used techniques are background subtraction and color filter. In our proposed approach, segmentation is performed using the second method. In each color space, the human skin color does not fall randomly, but clusters at a small area. There are two color spaces which are often used for locating skin region: HSV and YCrCb [10]. According to our experience, the HSV model is convenient for applying the filter on different skin colors, so it has a high level of generalization. However, YCrCb is appropriate with each “individual” skin color (e.g. yellow-color skin in Asia) [11]. Our experiment works mainly with the HSV model.

Proposed by [12], the human skin color is composed by two poles of color: red (blood) and yellow (melanin), with medium saturation. They also found that skin has a low texture amplitude. The skin color properties are essential information and effective in tracking the hand. Their skin filter was proposed as follows: each pixel (in RGB) is converted into log-component values \( I, R_g, \) and \( B_y \) using the following formulas:

\[
\begin{align*}
I &= L(G) \\
R_g &= L(R) - L(G) \\
B_y &= L(B) - \frac{L(G) + L(R)}{2}
\end{align*}
\]

where \( I, R_g \) and \( B_y \) are respectively log-components with color channels green, red (minus green) and blue (minus green and red). Intensity is represented via the green channel because of the poor spatial resolution of red and blue channels captured from some cameras. The constant 106 simply scales the output of the log function into the range \([0, 255]\). Value \( n \) is a random noise which prevents banding artifacts in dark areas, generated from a uniform distribution over the range \([0, 1]\). The constant 1 is added to prevent excessive inflation of color distinctions in very dark regions. The log transformation makes the \( R_g \) and \( B_y \) values independent of illumination level. Hue color at each pixel is determined based on \( \arctan(R_g, B_y) \):

\[
Hue = \frac{180}{\pi} \arctan^{-1}(R_g, B_y)
\]

The saturation value at each pixel is calculated as below:

\[
Saturation = \sqrt{R_g^2 + B_y^2}
\]

In order to obtain texture amplitude values, the intensity image is smoothed using a median filter. A subtraction between the original image and filtered
result is then performed. The absolute values of these differences are run through a second median filter. If a pixel falls into either of the following ranges, it is a potential skin pixel:

\[
\text{Texture} < 5, 110 \leq \text{Hue} \leq 150, 20 \leq \text{Saturation} \leq 60 \quad (7)
\]

\[
\text{Texture} < 5, 130 \leq \text{Hue} \leq 170, 30 \leq \text{Saturation} \leq 130 \quad (8)
\]

A skin detection result is illustrated in Fig. 2. Instead of performing skin filter on each frame, some tracking methods can be used to reduce the computational cost such as Camshift and particle filter [13]. This step was not necessary in this study.

### 3.1.2. Hand extraction

After applying the skin filter, the image often has a slight noise affecting the information contained within. Therefore, some morphological procedures which allow removing noise and smoothing the object boundary are performed. In some cases, images captured via the camera come with faces. The AdaBoost algorithm, a simple but powerful method, is combined with Haar-like features [14] to detect and locate the face region. In addition, a threshold value of object size may need to be define to ignore the large-size noise.

Many studies work with the covered arm, thus skin segmentation produces the separated hand, which can be used immediately for next processing stages. However, when such systems are applied in practice, the filtered hand consists of the arm, corresponding to the information that does not need identification. According to observations, we found that the wrist can be represented by a line segment, which two peaks are on concave regions of hand contour, as illustrated by two triangle-shape points in Fig. 2(c). Thus, the wrist’s position can be located based on this characteristic. The following steps are to determine the wrist and perform the arm removal. In addition, some obtained information is also used in feature extraction stage.

#### 3.1.2.1 Locating contour points

After applying the skin color filter, the silhouette of the hand is obtained. The contour needs to be determined to describe the hand shape and extract some other information. The algorithm presented in [15] is used to detect these points.

#### 3.1.2.2 Determining convex hull

Finding the convex hull of a shape is a well-known problem in the research field of computation geometry. The convex hull and concave contour usually contain most information of a 2D object. Let us imagine that this concept is similar to a rubber band which surrounds the hand, with some of contour points touch this band. The shape of this band is how the convex hull is looked like. The convex hull of the hand silhouette is determined based on the detected contour points using the three-coin algorithm [16]. In Fig. 2(c), the convex hull is represented by thin lines.

#### 3.1.2.3 Fingertip detection

Finger is considered as an important component which describes the hand state. Our approach uses the information related to fingertip positions for separating arm, and it is also needed for recognition process. In [17], the authors located fingertip based on all pixels of the hand contour. In our approach, their method is applied but to only some specific points on the contour to reduce the computational cost. Looking at Fig. 2(c), when the fingers are straightened, the convex hull always goes through the fingertips. Therefore, we only consider contour points, which are elements of the convex hull. At first we propose a simple strategy to cluster these points as follows (as illustrated in Fig. 3):

- Choose a point as the starting and label it
- List considered points in order based on the contour. A variable \( n \) is define as the number of these points.
- From the starting point, repeat this step \( (n - 1) \) times:
  - If the distance between the current and next points (the number of contour points

---

**Fig. 2:** Skin segmentation and locations of points of interest: (a) original frame, (b) hand silhouette, and (c) hand contour and 8 points of interest which need to be located for extracting features.

**Fig. 3:** Example of clustering points: the five white circles in the left image represent clusters which need to be determined; the right image is the illustration of contour points, which are on the convex hull, with corresponding clusters. \( P_1 \) is chosen as the starting point and is labeled \( L_1 \). The points of interest are listed in order as follows: \( P_1, P_2, P_3, ..., P_{27} \). The point \( P_1 \) is adjacent to \( P_2 \), therefore the label of \( P_2 \) is same as \( P_1 \) (\( L_1 \)). According to the left image, the distance between \( P_2 \) and \( P_3 \) is small, so that the label of \( P_3 \) is also \( L_1 \). \( P_3 \) is assigned a new label (\( L_2 \)) because there is a large concave region from \( P_1 \) to \( P_3 \). After labeling for all points, we see that the label which is set to \( P_{27} \) has to be reassigned as \( L_1 \).
between them) is less than a given threshold $h$, the next point is labeled same as the previous.

- Otherwise, the next point is assigned a new label.

Check the starting and ending points, if their distance is less than the define threshold $h$, the last label is set as the first label.

With each cluster, we apply the method in [17] to detect the candidate of fingertip. An angle value corresponding to each point is computed based on two related points, where the numbers of contour points between each of them and the point of interest are same (as illustrated in Fig. 4). Each fingertip is represented by only one single pixel. Its coordinate is located by choosing the point which is corresponding to the local maximum and greater than a given threshold.

3.1.2.4 Locating the deepest point of each concave contour

As described above, our objective is finding positions of the wrist. Looking at Fig. 2(c), it is not difficult to see the relation of the depth points (triangle and circle-shape points) with the convex hull. These depth points can be located by scanning each concave contour and finding the point has largest distance to the corresponding line of convex hull. In this approach, the Euclidean distance is used for computation. Now, at least one triangle-shape point needs to be determined for locating wrist.

3.1.2.5 Calculating palm centroid’s position

In Fig. 5(a), we see that the depth points surround the palm. Thus, the centroid is defined as the center of the smallest circle which covers all the depth points. This minimum enclosing circle is determined using the Skyum’s algorithm [18].

3.1.2.6 Arm detection and separation

As illustrated in Fig. 5(b), if the palm (minimum enclosing circle described above) is removed, obtained objects include finger and the arm. The area we need to locate, the arm, is the largest object which does not contain any fingertip point. The wrist is defined as the line goes through the depth point which is closest to the centroid of the arm and is perpendicular to the arm contour as shown in Fig. 5(c). By performing the separation at the wrist location, the silhouette of the hand-without-arm is obtained for next processing steps.

3.2. Feature extraction

In this stage, a set of attribute descriptions is created. The calculated values are stored as a vector. Our used features consist of general (ratio of width to height, wrist angle, motion orientation of hand center, the change of size and number of fingers and detailed information (calculated based on fingertip and cross sections). They are described in detail as follows:

3.2.1. Motion orientation of hand center

The necessity of this characteristic depends on each specific gesture. It is performed to ensure the general level of proposed approach. This feature, which provides the moving direction of the fist, is appropriate to represent some gestures such as hand-waving and punching. In addition, the calculation does not significantly affect the system performance because of its simplicity. The value of orientation is determined as follows, with the range is $[0, 2\pi)$:

\[
\text{Angle} = \begin{cases} 
\cos^{-1}\left(\frac{x'-x}{\sqrt{(x'-x)^2+(y'-y)^2}}\right) & y' \geq y \\
2\pi - \cos^{-1}\left(\frac{x'-x}{\sqrt{(x'-x)^2+(y'-y)^2}}\right) & y' < y
\end{cases}
\]

where $(x, y)$ and $(x', y')$ are two pairs of coordinate of the hand center in two consecutive frames.

3.2.2. Change of boundary size

In some gestures, the change of hand size is considered as the most important property, e.g. the up-down pattern in our experimented dataset [19]. It can be represented by the varying of bounding box, which is computed via the following equation:

\[
C = \sqrt{(w - w')^2 + (h - h')^2}
\]

where $(w, h)$ and $(w', h')$ are two pairs of the width and height values of the hand in two consecutive frames, respectively.
3.2.3. The number of fingers
This characteristic can be computed by counting finger tips which were detected in previous processing stage. It is a simple but powerful feature to reduce the risk of false identification.

3.2.4. Wrist angle
In hand extraction step, the line which represents wrist is determined. The angle between this line and horizontal direction is used as the representation of hand angle [see Fig. 6(a)].

3.2.5. Ratio of width to height
This feature is calculated after the hand is rotated with the rotation angle is $\alpha$, where $\alpha$ is the wrist angle computed above. This is illustrated visually in Fig. 6(a). The rotated hand is also used in extracting characteristics based on cross sections.

3.2.6. Angle of finger
This characteristic gives a set of five values corresponding to five possible fingers. They are calculated as the angles between wrist and lines which connect the palm centroid with fingertips. The default value is 0. When these values are put into the feature vector, they are sorted in descending order. In this way, the fingers are arranged in a certain order.

3.2.7. Cross sections
This feature represents the changes of pixel values through cross lines, which are split evenly on the rotated hand. More cross lines mean that longer information is extracted, but the complexity and storage capacity are increased. Thus, the number of them can be chosen in a flexible way to get the best feature quantity. Four image edges are padded background pixels to get more accuracy in calculating the change of pixel values. Fig. 6(b) illustrates the cross lines with corresponding values obtained by counting changes from the background to the foreground pixel and the contrary. In our approach, we use 5 horizontal and 5 vertical cross lines for describing the hand, thus we have 10 elements of the feature vector.

By combining 20 values described above, the feature vector which contains 20 elements is used in training and recognition.

3.3. Recognition
A dynamic gesture can be considered as a sequence of static gestures continuously over time. If the recognition is performed on each individual frame, the obtained accuracy is almost low. Therefore, the recognition has to be performed on these sequences. We use two well-known algorithm: DTW, which represents template matching methods, and HMM, a typical state space model, in order to compare their accuracies.

3.3.1. Hidden Markov Model
Each static gesture in a sequence is represented as a scalar by converting its feature vector into a codeword using k-means clustering method. In the other words, all vectors in feature space are clustered and each cluster is considered as a codeword. In our HMM, the number of observations is equal to the value of $k$. Each dynamic gesture is represented as a sequence of code words, which will be put into HMM as a feature vector. After training HMM using Baum Welch algorithm [20], the obtained model is combined with the coordinates of cluster’s centroids to recognize unknown gestures. According to experimental results, the value $k$ is assigned as 16, and the HMM has 4 states in order to achieve highest recognition accuracy.

3.3.2. Dynamic Time Warping
In computer vision, the DTW [21] is usually used to compute the optimal matching path between two signals. This is an appropriate method for recognizing human actions, which can be performed with different speeds. At first, DTW represents each dynamic gesture as a sequence of feature vectors. The similarity of the sequence, corresponding to the testing gesture, and each training sample is then calculated. The recognition result is determined based on the maximum of obtained values (corresponding to minimum distance between two sequences).

4. Experiment Results
Our system is implemented in C++ language using the Intel OpenCV library. The dynamic hand gesture dataset is SKIG which is collected from University of Sheffield [19]. This dataset contains 1080 video sequences from 6 people. It consists of 10 categories of hand gestures which are circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, character “Z”, cross, come here, turn around and pat. All these gestures were performed with three postures: fist index and flat. We randomly selected 540 samples for training. The testing results on 10 gesture types are shown in Fig. 7, in which the average recognition accuracies of DTW and HMM are 85.0% and 87.9%, respectively. The results show that the latter solution gives better recognition accuracy in
most of the ten gestures. Therefore, a specific method only gives best results with some certain gestures.

In addition, a comparison of recognition accuracies corresponding to different approaches are provided in Table 1. Most of the characteristics in our approach are extracted similarly to biological information of a hand such as the number of fingers angles between wrist and fingers, while other methods are performed based on pixel-level information. With each hand gesture, the variation of features corresponding to different samples in proposed approach is less than other solutions. Therefore, the stability of our approach is higher than others, leading to the superiority of corresponding experimental results.

When testing our system with images captured directly via a web camera, the processing time for each frame is in the range of 25 - 32 milliseconds. The used camera is Logitech QuickCam Sphere with frame-per-second is 30 and resolution is 640 × 480. It shows that our proposed approach is appropriate for developing a real-time system of dynamic hand gesture recognition.

5. Conclusion and Discussion

In this paper, an approach with a low computational cost is proposed to recognize dynamic hand gestures.

Our system consists of three following stages: preprocessing, features extraction and recognition. At first the segmentation is performed based on color information to extract hand silhouette. The possible arm is then separated and removed using geometrical properties. In second stage, these attributes are used to compute the characteristics, which describe the general and detailed information of the hand. They are robust features since they can exactly represent the information of hand shape, which greatly affects the gesture recognition problem, without depending on the angle of inclination. Finally, the DTW and the combination of k-means clustering method and HMM are performed for recognizing gestures in order to give a comparison of accuracies. The experimental result shows that the described features can represent the hand state in a sequence of static gestures, including shape, orientation and motion. Besides, our system can be developed for executing in real time. In addition, features which are described in our approach can be used as a foundation for recognizing static hand gestures and similar problems.

As further work, our method will be improved for specific problems, such as sign language identification and human-computer interaction. The YCrCb color space will be studied for skin segmentation. Based on the visual distributions of skin values [11], we will try to approximate an ellipse around each area. The determined parameters of such ellipses are used to classify pixels. The features extracted from contours (inside the hand) will be also considered to increase the level of detail of hand representation. In addition, performing the recognition with other machine learning algorithms is needed to give a general comparison, and the most suitable model can be chosen for each problem.

Acknowledgment

The authors would like to thank Trong-Nguyen Nguyen, University of Montreal, Canada. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References


Duc-Hoang Vo received the B.S. degree in Information Technology from the University of Science and Technology, Danang, Vietnam, in 2006, the M.Sc. degree in Computer Science from the University of Danang, in 2011. He is now a PhD student and also a lecturer at University of Science and Technology, Danang, Vietnam. His current research field is pattern recognition.

Huu-Hung Huynh received the B.S. degree in Information Technology from the Hanoi University of Technology, Vietnam, in 1998, the M.Sc. degree in Computer Science in 2003, and the Ph.D. degree in Computer Science from the Aix-Marseille University in 2010. He now is lecturer at University of Science and Technology, Danang, Vietnam. His current research interests include computer vision and its applications to medical imaging and health care.

Jean Meunier received the B.S. degree in physics from the Universit de Montreal, Canada, in 1981, the M.Sc.A. degree in applied mathematics in 1983, and the Ph.D. degree in biomedical engineering from the cole Polytechnique de Montral, Canada, in 1989. In 1989, after postdoctoral studies with the Montreal Heart Institute, he joined the Department of Computer Science and Operations Research, Universit de Montral, where he is currently a Full Professor. He is also a regular member of the Biomedical Engineering Institute at the same institution. His current research interests include computer vision and its applications to medical imaging and health care.