Feature selection for visual gesture recognition using hidden Markov models

Abstract

Hidden Markov models have become the preferred technique for visual recognition of human gestures. However, the recognition rate depends on the set of visual features used, and also on the number of states of the hidden variable. It is difficult to determine a priori the optimal set of features and number of states. In this paper we analyse experimentally the use of different features for gesture recognition in an office environment. We considered a set of seven gestures that include interaction with other objects, such as writing, using the mouse, opening a drawer, etc. We use a single camera to detect and track the hand of the user based on adaptive colour histograms. From tracking the hand in a video sequence we obtain several features. The features considered include position and velocity in polar and Cartesian coordinates, and the trajectory represented as a chain code. Given that these features are continuous, we discretized them into a set of symbols using vector quantization. We then tested the recognition rate using HMMs with different: (i) number of discrete symbols, (ii) number of hidden states, (iii) combination of features. The results show a high variation on the recognition rate depending on these parameters, from below 50% to more than 95%. The best performance (97%) was obtained by using the magnitude and orientation in polar coordinates, 64 discrete symbols and 10 states.

1. Introduction

The detection and understanding of human gestures in videos is of high value for many applications, such as human-computer interaction, surveillance, collaborative environments, training and entertainment, and medical support systems. Human gestures are characterized by the spatio-temporal structure of their motion pattern. These structures are intrinsically probabilistic and often ambiguous. In general, they can be treated as temporal trajectories in a high dimensional feature space representing closely correlated measurements on visual observations. For example, the spatio-temporal structure of a simple behavior such as moving the hand toward a telephone, could be represented by the trajectory of an observation vector given by the position and displacement of the hand (Figure 1).

Hidden Markov Models (HMMs) are widely used for modeling temporal structures. They have been applied to speech recognition [12], learning object movement and behavior models [7], and more recently gesture recognition [16], [1]. However, the recognition rate depends on the set of visual features used, and also on the number of states of the hidden variable. It is difficult to determine a priori the optimal set of features and number of states. The extraction of relevant visual information from an images sequence, and interpretation of this information for the purpose of recognition and learning is a challenging task. An adequate choice of visual features is very important for the success of gesture recognition systems [2]. The first attempts to extract relevant features to solve the gesture recognition problem, was the use of mechanical devices connected directly to the human body, to obtain hand and arm joint angles and spatial position [11], using glove-based devices [4]. Glove-based gesture interfaces require the user to wear a cumbersome device, and carry a lot of cables that connect the device and the computer, these are called intrusive methods. For more natural interfaces in gestures recognition systems, the features must be extracted from visual images without

Figure 1. Trajectory described by the hand in an interaction with surrounding objects in an office environment.
any external devices.

Current, non-intrusive approaches, use a single camera (or a stereo system) to extract relevant features while detecting and tracking the hands, or other parts of the body of a person; which are used as input to a recognition system. To build representations that accurately capture the activities or gestures taking place, image features of people and objects must be carefully identified and extracted. A sequence of extracted features should be able to construct a meaningful representation of the gestures. The most commonly used methods for feature representation are: trajectory-based features, optical flow, and region-based features [14]. However, the choice is closely related to application and environmental conditions.

In this study, we use different trajectory-based features (position, velocity, direction and orientation) and analyze experimentally their impact on the recognition rate for a gesture recognition system in an office environment. We focus our attention on human gestures performed by a hand that include interaction with other objects. These actions include: picking up the telephone, writing, using the mouse, opening a drawer, etc. Given that the features are continuous, we discretized them into a set of symbols using vector quantization. We then tested the recognition rate using HMMs with different: (i) number of discrete symbols, (ii) number of hidden states, (iii) combination of features. The results show a high variation on the recognition rate depending on these parameters, from below 50% to more than 95%. The best performance (97%) was obtained by using the magnitude and orientation in polar coordinates, 64 discrete symbols and 10 states.

The rest of the paper is organized as follows. In Section 2, we describe hand localization and tracking using color-based and adaptive histograms techniques. Section 3 describes the feature extraction process. In Section 4 we present the learning and recognition process using HMMs. The experimental results and analysis are in Section 5. Related work is shown in Section 6, and in Section 7 we give conclusions and directions for future work.

2. Hand detection and tracking

For this work, we consider 7 types of gestures performed in an office environment. Hand gestures are made on a planar space. The view of the scene is provided by a downward pointing, ceiling-mounted camera which offers several advantages for hand and object tracking, such as a less obstructed perspective of the gestures. Our hand detection and tracking approach is divided into two stages. The first step is the hand localization process, that obtains the hand region from an image using and adaptive color histogram. The second step describes the hand tracking algorithm, that generates the gesture trajectory by connecting the hand centroid along the continuous time sequence.

2.1. Hand detection

For detection of the hand region in an image we use a color-based approach [8]. Skin color is usually more distinctive and less sensitive to illumination changes if we use the YIQ color space [5]. In this study we use the YIQ color coordinate system. In order to reduce the effect of lighting, only the values I and Q of the pixels are used. The skin color histogram model M can be defined as follows:

\[ M = (i, q, count) \]

where \(i, q\) range over all the possible values of I and Q.

Color histogram matching is the technique used to detect the regions covered with a specific color. Each window in the input image is matched to the color distribution of the skin color model histogram. The result of matching is represented by a match score, which indicates the degree of the color distribution similarity between the input image (or window) and model color histograms. The similarity is computed by:

\[ C_k = (i_k, q_k, counter_k), \quad (1) \]

\[ Similarity_k = \frac{\sum_{i=i_k-q_k}^{i_k-q_k} \min(counter, counter_k)}{A}, \quad (2) \]

where \(C_k\) is the \(I-Q\) color histogram of \(k\)th window, \(Similarity_k\) is the similarity value \([0.0, 1.0]\) of the \(k\)th window, and \(A\) is the number of pixels in a window. This method, also known as histogram intersection, was proposed by Swain and Ballard [15]. The skin color similarity is used to distinguish between skin and non-skin regions in an image. Swain and Ballard used a fixed threshold value. However, it is difficult to adapt to changes in lighting conditions when a fixed threshold value is used to binarize the similarity image. Our system uses an adaptive thresholding method, where the threshold values vary with each similarity image. The method proposed by Otsu [10] is used to determine the threshold value. This method chooses an adaptive threshold value that maximizes the variance between the two groups, skin and non-skin regions.

Otsu Algorithm. Otsu proposed an algorithm for automatic threshold selection from a histogram of an image. Let the pixels of a given image be represented by \(L\) gray levels \([1, 2, ..., L]\). The number of pixels at level \(i\) is denoted by \(n_i\), and the total number of pixels by \(N = n_1 + n_2 + ... + n_l\). Then suppose that the pixels were dichotomized into two classes \(C_0\) and \(C_1\), which denote pixels with levels \([1, ..., k]\) and \([k + 1, ..., L]\), respectively. This method is based on a discriminant criteria, which is the ratio of between-class variance and total variance of gray levels.
2.2. Hand Tracking

Once that skin regions have been detected in an image sequence, the next step consists on performing hand tracking. For hand tracking, we have to decide if regions with labeled skin pixels in an image are the face or hand of a person. Hand detection is based on two rules. The first rule assumes that only the hand and face of the person cause a significant movement in the images sequence. The second rule, establishes a minimum threshold (number of skin labeled pixels) that a region must have to be considered the hand or face of a person (values obtained from $C_k$). Once determined if a labeled skin region corresponds to the hand of a person, we obtained the center points. A center point of an object may be defined using the centroid of the object $(X_c, Y_c)$:

$$X_c = \frac{\sum_x \sum_y B(x,y)x}{A}, \quad Y_c = \frac{\sum_x \sum_y B(x,y)y}{A}$$

(3)

where $A$ is the number of pixels in the object and $B$ is the binarized input object which takes two values, 1 for the hand and 0 for the background. Hand tracking is realized by applying the hand detection process over a search window based on motion heuristics (maximum motion between frames) over the image sequence. In the time domain, the sequences of centroid points are detected by the hand localization algorithm, and thus, the gesture trajectory $G$ is produced by connecting centroid points.

$$G = (x_1, y_1), \ldots, (x_n, y_n)$$

(4)

Figure 2 shows the hand detection and tracking process. The system detects the face and right hand of a person that interacts with objects. We establish the origin as the coordinates of the face of the person to obtain relative information between the face and other parts of the body. Although at the moment we do not use the relative position of the face and other objects, we plan to use it in the future as a context to improve gesture recognition.

3. Feature Extraction

The success of a general recognition system depends on an adequate set of features representing the patterns. In previous work, gesture recognition was performed in terms of a standard $x - y$ coordinate system. Under this coordinate system, there are several methods for representing a gesture trajectory, such as using raw position, Cartesian velocity, polar velocity and angular velocity, which was proposed by Campbell [2]. There are many other types of features such as chain code, mesh code, momentum and so on. Yoon et al [16] show than all of the features are based on only three basic characteristics of a gesture trajectory: (i) distance to the origin ($\rho$), (ii) angle ($\theta$), and (iii) velocity ($v$). In this coordinate system, the feature sets are affected by changes in the size and angle of the input gesture trajectory. So it is necessary to normalize these basic features. To reduce normalization error and computing time, we can transfer from the Cartesian system to a polar system. Under the polar system, the feature sets are robust to and independent of size and angle variations.

3.1. Feature Selection

In this study we use the following features: i) the set of basic features proposed by Yoon, with those features we can characterize the motion pattern described by the hand during a gesture and, ii) a chain code. Chain codes [5] is a technique used for obtaining the geometrical structure (shape) of objects. It is based on a fixed grid with a fixed set of possible orientations. Since the grid is normally uniform, direction is sufficient to characterize displacement. We used a set of 16, 32, and 64 directions numbers as chain code elements, to represent the shape of the trajectory of a hand gesture. We consider that these features can characterize the type of gestures that we intend to recognize.

The three basic features ($\rho-\theta-v$) proposed by Yoon, are integrated by position information under the polar system, and velocity information in the Cartesian system. Therefore, these features are independent of variance of translation, rotation and scaling of a gesture trajectory. To obtain these features, the following equation is used:

$$(C_x, C_y) = \left(\frac{1}{n} \sum_{t=1}^{n} X_t, \frac{1}{n} \sum_{t=1}^{n} Y_t\right)$$

(5)
where \((C_x, C_y)\) means the center point of a gesture trajectory. Then each gesture has a different center point (centroid).

\(\theta_i\) is the angle obtained between each gesture point and the center point (centroid) of all points that describes the gesture trajectory:

\[
\theta_i = \tan^{-1} \left( \frac{Y_i - C_y}{X_i - C_x} \right)
\]

\(r_i\) is the distance (magnitude) between the center point of the gesture trajectory and each single point:

\[
r_i = \sqrt{(X_i - C_x)^2 + (Y_i - C_y)^2}
\]

\(r_{\text{max}}\) is the largest distance from a center point to any point in one gesture:

\[
r_{\text{max}} = \max_{i=1}^{n} (r_i)
\]

\(\rho_i\) is a normalized value between 0.0 – 1.0. The \(\rho_i\) value is the result of size normalization. The \(\phi_i\) value is the result of angle normalization:

\[
\rho_i \frac{r_i}{r_{\text{max}}}, \phi_i = \frac{\theta_i}{2\pi}
\]

The factor \(v_i\) is for velocity between 2 points and also normalized between 0.0 – 1.0:

\[
V_i = \sqrt{(X_i - X_{i+1})^2 + (Y_i - Y_{i+1})^2}
\]

\[
V_{\text{max}} = \max_{i=1}^{n} (V_i)
\]

\[
v_i = \frac{V_i}{V_{\text{max}}}
\]

Finally, \(F_1\), represents the trajectory of a gesture in the \(\rho - \phi - v\) coordinate system:

\[
F_1 = \{(\rho_1, \phi_1, v_1), \ldots, (\rho_n, \phi_n, v_n)\}
\] (6)

\(F_2\), represents information of the gesture trajectory given by chain code elements:

\[
F_2 = \{\alpha_1 = \text{deg} 0, \alpha_2 = \text{deg} 15, \ldots\}
\] (7)

The values of chain code are obtained between two consecutive points, considering the normalized gesture trajectory.

3.2. Features Discretization and Vector Quantization

In this paper we use discrete hidden Markov models. Thus, the features are quantized to obtain discrete symbols as observations for the HMM.

The chain code description of the trajectory is discretized by considering a fixed number of directions in the representation. Once the gesture trajectory has been normalized, then we obtain vectors with direction numbers representing the chain codes, in sizes of 16, 32 and 64, respectively.

For the gesture trajectory in the \(\rho – \phi – v\) coordinate system, we obtain discrete symbols by using the \(k\)-means vector quantization algorithm [6]. The \(k\)-means algorithm is an efficient and simple technique for clustering. The algorithm is based on the minimum distance between the center point of each cluster and feature points. \(k\)-means performs the following steps:

Step 1. Initialization. Build up an initial VQ codebook \((x_i, 1 \leq i \leq k)\).

Step 2. Repeat the following for all training vectors:

Classification. Classify each vector \(x_k\) into one of the clusters \(C_i\) using the minimum distance.

Codebook updating. Update the codebook by computing the centroid of the training vectors in each cluster.

Step 3. Termination. Terminate if the decrease in the overall difference value is below a threshold value; otherwise go to Step 2.

The codebook generated consists of the symbol number and the coordinates of the centroid of each cluster. In the experiments, the symbol code of an observation can be determined using the distance between an observation and the centroid of each cluster. We used different number of clusters in the experiments. We tested with codebooks of size 32, 64 and 128. In figure 3 (see results section) we observe the impact of varying the size of the codebook for the case where we use the 3 basic Yoon features. In this case we obtained the best recognition score with a codebook of 64.

We compare experimentally the recognition rate of the gestures recognition system based on HMMs. We consider three aspects that could affect the recognition rate: a) the number of hidden states and, b) the feature set, c) the number of symbols (discretization). In the following section we review HMMs and then we present the experiments.

4. Hidden Markov Models

One reason for the popularity of the HMM has been their ability to accurately characterize data exhibiting sequential structure in the presence of noise, such as in speech and gestures. A HMM has the ability to find the most likely sequence of states that may have produced a given sequence of observations. Formally, the elements of a hidden Markov model are defined using the following declarations:

- set of observation strings \(O = O_1, \ldots O_t, \ldots, O_T\),
  
where \(t = 1, \ldots, T\)
• set of $N$ states $S_1, ..., S_N$
• set of $k$ discrete symbols from a finite alphabet $V_1, ..., V_k$
• a state transition matrix $A = a_{ij}$, where $a_{ij}$ is the transition probability from state $S_i$ to $S_j$
• an observation probability matrix $B = b_{jk}$, where $b_{jk}$, is the probability of generating symbol $V_k$ from state $q_j$
• the initial probability distribution for the states

$$\prod = \pi_j, j = 1, 2, ..., N; \pi_j = Pr(S_j; ent = 1)$$

The complete parameter set of a HMM can be expressed compactly as $\lambda = (A, B, \pi)$. Three basic problems must be solved for the application of HMMs: evaluation (classification), decoding, and training [12]. We approach the above problems with the following techniques: forward algorithm, Viterbi algorithm, and the Baum-Welch algorithm.

The HMM topology used in this paper is the classical left–right (Bakis) structure, which is typical for motion ordered paths, such as gestures we are trying to recognize.

5. Experimental Results

In this study, we focus our attention on human gestures performed by a hand that include interaction with other objects. The type of gestures considered in the experiments are the following: pick up telephone, erasing, writing, using the mouse, opening the desk drawer and, printer on.

5.1. Training

A data base with 4200 data points was generated for training and testing. Data stored in the DB was obtained from 100 gestures sequences with 7 different gestures realized by a person in a sitting position and interacting with objects in an office environment. Gestures were captured by a fixed ceiling mounted camera pointed downward. We used 2100 data points during the HMMs training phase and 2100 data during the testing phase.

5.2. Results

The following sets of features were used in the experiments:

• polar magnitude-polar orientation-Cartesian velocity (Figure 3),
• chain-code or quantized direction vector (Figure 4),

Figure 3. Results for the 3 basic features: polar magnitude and orientation, and Cartesian velocity, for different number of states and codebook sizes.

Figure 4. Results with chain code for different number of directions.

• polar magnitude (Figure 5.a),
• Cartesian velocity (Figure 5.b),
• polar orientation-Cartesian velocity (Figure 6.a),
• polar magnitude-polar orientation (Figure 6.b),
• polar magnitude-Cartesian velocity (Figure 6.c).

For each case, we vary the size of the codebook (VQ) and the number of hidden states. The results of these combinations are summarized in table 1. For all the cases, the recognition rate is the mean value for the test sequences.

Figure 3 shows the results when we are using the combination of three basic features (polar magnitude-polar orientation-Cartesian velocity) proposed by Yoon. The best rate recognition (88%), corresponds to a 64 codebook size, and 7 hidden states. Figure 4 shows the recognition rate when we use the chain-code as the only feature. The best rank in this case (96%) corresponds to 32 directions, and 7 hidden states. We observe that using this feature we obtain a high recognition rate, although this is dependent on
the number of states. The chain code feature has been used by others, obtaining a similar performance [3]. Cartesian velocity is not a adequate choice of feature in this case. Figure 5.b shows a poor performance, the recognition rate varies from 25% to 75%. A possible cause is that the hand velocities for different gestures are very similar. Figure 6 shows the recognition rate when we are using the combination of two features. The combination of polar orientation and Cartesian velocity is shown in Figure 6.a, here we observe a stable behavior in the recognition rate with a 64 codebook size and obtaining the best score (93%) with 9 hidden states. The combination of polar magnitude-polar orientation, reports the best recognition rate of all, with a best score of (97%), with a 64 codebook size and 10 hidden states. Then, the use of normalized polar coordinates applied with trajectory features, generates good performance in our system.

Under appropriate features selection, HMMs can efficiently characterize motion profiles. To use many features not always corresponds in obtaining better results, as demonstrated by these results. We observe that trajectory-based features, using the polar coordinate system and normalized data, generate a good performance. We observe that the recognition rate is also dependent on the number of states and observation symbols.

5.3. Implementation

The system was implemented over a personal computer, Intel Pentium 4, 1.3 Ghz processor. Pixel-view is the video capture card, this card allow us to capture 30 frames per second, we used 320x240 pixels images. Images were processed between 12-15 fps. Sony TRV19 CCD color is the video camara used, the parameters were manually tuned. Software was implemented using Visual C++ 6.0, over Win-
Table 1. Recognition rates for different features combinations and number of states.

<table>
<thead>
<tr>
<th>States</th>
<th>Features</th>
<th>ρ</th>
<th>v</th>
<th>ρv</th>
<th>ρσ</th>
<th>ρvσ</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>87</td>
<td>96</td>
<td>66</td>
<td>50</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>87</td>
<td>96</td>
<td>63</td>
<td>51</td>
<td>91</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>88</td>
<td>96</td>
<td>72</td>
<td>51</td>
<td>92</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>86</td>
<td>90</td>
<td>70</td>
<td>60</td>
<td>92</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>84</td>
<td>91</td>
<td>69</td>
<td>79</td>
<td>93</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>82</td>
<td>88</td>
<td>68</td>
<td>78</td>
<td>92</td>
</tr>
</tbody>
</table>

6. Related work

The task of gestures recognition is a very challenging problem in computer vision. Several recognition models have been proposed, HMMs seem to be one of the preferred approaches. However, a critical aspect for good performance is the selection of features. This means, finding the smallest and richest feature set that will allow us to properly characterize human-based motion (specifically hand-based motion).

Many approaches have used with different feature vectors. Aviles used [1] trajectory-based features, position and direction in Cartesian coordinates. Campbell et al. [2] uses functions of different coordinates (Cartesian, polar, angular) as feature vectors, and applies these in the recognition of six Ti Chi movements. Joint arm angles have been used by Quan as feature vectors for recognition of human activities [11]. The sequence of points positions, \((x, y)\), in the trajectory described by the centroid of the hand during an activity has been used by Darnell [9] and Rao [13] as features vector. Yoon has applied a simplified feature vector formed by: normalized polar orientation and magnitude, and Cartesian velocity [16].

Although many different features have been applied for gesture recognition, there are few systematic studies to compare them empirically. This is important given the high variation on recognition rates for different feature sets, and its dependence on other parameters such as the number of states and observation symbols.

7 Conclusions

In this paper we analyze experimentally the use of different features for gesture recognition using HMMs. We considered a set of hand gestures that include interaction with other objects in an office environment. We use a single camera to detect and track the hand of the user based on adaptive colour histograms. From tracking the hand in a video sequence we obtain several features. The features considered include position and velocity in polar and Cartesian coordinates, and the trajectory represented as a chain code. Given that these features are continuous, we discretized them into a set of symbols using vector quantization. We then tested the recognition rate using HMMs with different: (i) number of discrete symbols, (ii) number of hidden states, (iii) combination of features. The results show a high variation on the recognition rate depending on these parameters, from below 50% to more than 95%. The best performance (97%) was obtained by using the magnitude and orientation in polar coordinates, 64 discrete symbols and 10 states. We concluded that the performance of HMMs is highly dependent on the set of features and also in other parameters as the number of hidden states and observation symbols.

We believe that context information, such as the relation between the hands and other objects in the environment, provides another important set of features to improve gesture recognition. So in the future we will investigate the integration of context features together with more expressive recognition models for gesture recognition systems.

References


