Head Gesture Recognition Based on LK Algorithm and GentleBoost

Jianzheng LIU, Chang LIU, Zheng ZHAO

1. School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
2. SEIT, Tianjin Foreign Studies University, Tianjin 300204, China
jz_leo@163.com, liuchang2@tjfsu.edu.cn, zhengzh@tju.edu.cn

Abstract

Different head movements can help us not only understand accurately whether the speaker in question is in favor of or against the current topic but also recognize human expressions better, i.e. distinguishing similar expressions such as smile and scorn. Since the movements of the feature point -- nostrils on the face and the movements of the head are almost the same, tracing the feature point with Lucas-Kanade(LK) algorithm we can find out the pattern of the head movements. We can train GentleBoost classifiers use the difference of coordinates of nostril in an image or a frame to identify the movement of the head, include nodding, shaking, bowing and turning aside. A number of tests have been conducted and the results have proved the accuracy, reliability and efficiency of the method of discussed in the paper.

Keywords: LK Algorithm, Gentleboost, Head Movement, Head Gesture

1. Introduction

Human communication can be divided into verbal communication and non-verbal communication. Emotions is part of non-verbal communication. People tend to nod in agreement, but when a person agrees upon something that he does not actually agree, he tends to shake his head unconsciously. This behavior can be called micro-expressions. It is difficult to identify some similar expression such as smile and scorn when we recognize human facial expressions through the use of some facial expression identification system. These expressions are easy to distinct if we refer to human's head movement. A scorn expression is often accompanied by a shaking head.

There is a very classical approach to identify the movement of human body. Bobick and Davis[1] transformed image sequence into motion energy image (MEI) and motion history image (MHI), which MEI reflects the range and strength of motion, and MHI reflects the changes of motion in time. This method is very efficient but is not robust enough.

Based on the recognition and tracking eye area, there are a lot of methods that can be used in the recognition of human head gestures, such as [2-5]. But all the method mentioned above have various problems. Some of them cannot work in real-time, others are not robust enough. Some methods even need special equipment in their application, such as a IR camera to positioning and tracking pupils by highlighting them in images. There is still a common problem in the methods mentioned above, that all the methods can only identify head nodding and shaking, but not bowing and turning aside.

There are many features composed by human facial organs. According to some feature points took out certain parts of ROI(Region of Interest) from human face images, we can identify human expressions and head movement. Actually, except eyes, there are several feature points that never move with the change of facial expressions (such as nostrils). When a person nods, these points will move up and down in images. When a person is shaking his head, the points will move around.

So we proposed an robust approach which can identify human head movement quickly and automatically. The method is based on LK algorithm and Boost algorithms that can automatically locate and track nostril and identify the movement of the head. Our method can identify not only nod and shake, but also bow head and turn face aside. Experiments show that the method can identify human head movement quickly and actually.
2. Related Works

2.1. LK algorithm

We base our solution on a previous result by Tomasi and Kanade [6], who proposed a method to solve the problem. In general, any function of three variables \( I(x, y, t) \), where the space variables \( x \) and \( y \) as well as the time variable \( t \) are discrete and suitably bounded, can represent and image sequence. Images taken at near time instants are usually strongly related to each other. We usually express this correlation by saying that there are patterns that move in an image stream. This means that the function \( I(x, y, t) \) is not arbitrary, but satisfies the following property:

\[
I(x, y, t + \tau) = I(x - \xi, y - \eta, t). \tag{1}
\]

A later image taken at time \( t + \tau \) can be obtained by moving every point in the current image, taken at time \( t \), by a suitable amount. The amount of motion \( d = (\xi, \eta) \) is called the displacement of the point at \( x = (x; y) \) between time instants \( t \) and \( t + \tau \).

If we redefine \( J(x) = I(x, y, t + \tau) \), and \( I(x-d) = I(x - \xi, y - \eta, t) \), where the time variable has been dropped for brevity, our local image model is

\[
J(x) = I(x-d) + n(x), \tag{2}
\]

where \( n \) is noise.

The displacement vector \( d \) is then chosen so as to minimize the residue error defined by the following double integral over the given window \( W \):

\[
\varepsilon = \int_W \left[ I(x-d) - J(x) \right]^2 wdx. \tag{3}
\]

In this expression, \( \omega \) is a weighting function.

When the displacement vector is small, the intensity function can be approximated by its Taylor series truncated to the linear term, and we can write the residue defined in (3) as

\[
\varepsilon = \int_W \left[ I(x) - g^T d - J(x) \right]^2 wdx = \int_W (h - g^T d)^2 wdx \tag{4}
\]

where \( h = I(x) - J(x) \).

This residue is a quadratic function of the displacement \( d \). As a consequence, the minimization can be done in closed form. Differentiating the last expression of the residue \( \varepsilon \) in (4) with respect to \( d \) and setting the result equal to zero yields the following vector equation:

\[
Gd = e \tag{5}
\]

where the coefficient matrix is the symmetric, \( 2 \times 2 \) matrix

\[
G = \int_W gg^T wdx = \int_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} wdx = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{bmatrix}
\]

Head Gesture Recognition Based on LK Algorithm and GentleBoost

Jianzheng LIU, Chang LIU, Zheng ZHAO
and the right-hand side is the two-dimensional vector
\[ e = \iint_W (I - J) g w dA \]

Equation (5) is the basic step of the tracking procedure. For every pair of adjacent frames, the matrix \( G \) can be computed from one frame, by estimating gradients and computing their second order moments. The vector \( e \), on the other hand, can be computed from the difference between the two frames, along with the gradient computed above. The displacement \( d \) is then the solution of system (5).

2.2. Feature Point

Actually, not all parts of an image contain motion information. We called a small window (3×3, 5×5, or 7×7 pixels, and so on) in an ROI as a "point". Most of the points in image is no use to us, but how can we select the point? A point we interested must be a window has a corner, we called it as a "feature point".

This means that the 2 × 2 coefficient matrix \( G \) of the system must be both above the image noise level and well conditioned[7]. In turn, the noise requirement implies that both eigenvalues of \( G \) must be large, while two large eigenvalues can represent corners, salt-and-pepper textures, or any other pattern that can be tracked reliably.

As a consequence, if the two eigenvalues of \( G \) are \( \lambda_1 \) and \( \lambda_2 \), we accept a window if

\[ \min(\lambda_1, \lambda_2) > \lambda \] (6)

where \( \lambda \) is a predefined threshold.

To determine \( \lambda \), we first measure the eigenvalues for images of a region of approximately uniform brightness, taken with the camera to be used during tracking. This gives us a lower bound for \( \lambda \). We then select a set of various types of features, such as corners and highly textured regions, to obtain an upper bound for \( \lambda \).

2.3. Extract Features from Feature Point

Recent work [8] has shown that a Gabor approach for local feature extraction outperformed PCA (Principal Component Analysis), FLD (Fisher’s Linear Discriminant) and LFA (Local Feature Analysis). Nowadays, lots of researchers use Gabor filters to extract image features[9,10]. The essence of the success of Gabor filters is that they remove most of the variability in image due to variation in lighting and contrast, at the same time being robust against small shift and deformation. Gabor wavelets seem to be a good approximation to the sensitivity profiles of neurons found in visual cortex of higher vertebrates [11].

A 2D Gabor filter \( \psi(x,y) \) can be defined as:

\[ \psi(x,y) = \frac{\alpha \beta}{\pi} e^{-(\alpha^2 x^2 + \beta^2 y^2)} e^{i2\pi f_0 x} \] (7)

\[ x' = x \cos \theta + y \sin \theta \]

\[ y' = -x \sin \theta + y \cos \theta \]

where \( f_0 \) is the central frequency of a sinusoidal plane wave, \( \theta \) is the anti-clockwise rotation of the Gaussian and the plane wave, and \( \alpha \) and \( \beta \) are the parameters for scaling two axis of the elliptic Gaussian envelope. Here we consider that the orientation of the Gaussian envelope and the orientation of the sinusoidal function are the same (which is one the characteristics of complex cells of the mammals’ visual cortex). The Gabor function is actually Gaussian shaped function (first part of (7)) which modulates sinusoidal plane wave carrier (second part of (1)). Its 2D Fourier transform is:
$$\psi(u,v) = e^{-\pi i \frac{(u \cos \theta + v \sin \theta - f_0)^2}{\alpha^2}, \frac{(u \sin \theta - v \cos \theta)^2}{\beta^2}}$$ \hspace{1cm} (8)

By fixing the ratio of the frequency of the wave and the sharpness of the Gaussian we get that the spatial filter (7) includes a constant number of waves. The ratios which are known to hold for the cells in human visual cortex are [11]:

$$\gamma = \frac{f_0}{\alpha} = \frac{1}{\sqrt{\pi} \times 0.9025}$$ \hspace{1cm} (9)

$$\eta = \frac{f_0}{\beta} = \frac{1}{\sqrt{\pi} \times 0.58695}$$ \hspace{1cm} (10)

Thus, a normalized filter can be presented in spatial domain as:

$$\psi(x,y) = \frac{f_0^2}{\pi \gamma \eta} e^{-\frac{f_0^2 x^2}{\gamma^2}, \frac{f_0^2 y^2}{\eta^2}} e^{i2\pi f_0 x}$$ \hspace{1cm} (11)

and in frequency domain as:

$$\Psi(u,v) = e^{-\frac{f_0^2 (u \cos \theta + v \sin \theta - f_0)^2 + (u \sin \theta - v \cos \theta)^2}{\gamma^2 \eta^2}}$$ \hspace{1cm} (12)

Several Gabor filters are combined to form a filter pool, we called it as a "Gabor-pool". The pool is usually composed of filters in several orientations and frequencies, with equal orientation spacing and octave frequency spacing, while the relative widths of Gaussian envelope $\gamma$ and $\eta$ stay constant. In the frequency domain Gabor filter must obey Nyquist rule, which means that $f_0 \leq 0.5$, for each $\theta$.

2.4. Classifier and GentleBoost

In 1990, Schapir proposed an algorithm, which is the original Boosting algorithm [13], and later in 1995, Freund and Schapire have also proposed an improved AdaBoost algorithm. Our method used GentleBoost, an improved AdaBoost algorithm, which is faster than the convergence of AdaBoost, and for better implementation of object detection.
In contrast to AdaBoost, GentleBoost uses real valued features. GentleBoost [14] seems to converge faster than AdaBoost, and performs better for object detection problems. The performance of boosting methods on data which are generated by classes that have a significant overlap, in other words, classification problem where even the Bayes optimal prediction rule has a significant error is discussed in [15]. For this case, GentleBoost performs better than AdaBoost since AdaBoost over-emphasizes the atypical examples which eventually results in inferior rules.

The outline of the GentleBoost algorithm is as follows. At each boosting round, a regression function is fitted (by weighted least-squared error) to each feature in the training set. The fitting of the regression function is done for one feature for all training examples by minimizing the weighted error

$$\text{error}_W = \frac{\sum_i (w_i y_i - \alpha_i (x_i > \theta_i) + b_i)^2}{\sum_i w_i}$$

(14)

where i is the i-th training example. By minimizing weighted error through all features, we get the feature with the smallest error and with the adequate parameters which minimize this error (a, b and $\theta$). Next step is estimation of the fitting function $f_m$ for each training example with these parameters:

$$f_{m_i} = (a(x_i (\text{Feature Index}) > \theta) + b)$$

(15)

where FeatureIndex is the feature which is chosen in the round m. the next step is to update the classifier output and the weights for each training example:

$$F(x_i) = F(x_i) + f_{m_i}$$

(16)

$$w_i = w_i e^{-\gamma f_m}$$

(17)

Finally, the weights should be renormalized and for each testing example $x_i$ the output of the classifier should be calculated as:

$$\text{sign}[F(x_i)] = \text{sign} \left[ \sum_{m=1}^{M} f_m(x_i) \right]$$

(18)

where M is the number of the most relevant features the classifier has chosen for the classification.

3. Proposed Head Movement Recognition

As the previous section discusses, we can extract feature points from the area which never move with the change of expressions in human face. We can identify head movement in tracking these points. We tested a lot of feature points in our laboratory. We found that there are almost no difference between the images of different people, regardless of their age, sex and race. Pictures in the first row in Figure 1 are images of different people, and the images in the second row are their nostrils area. Figure 1 shows similarities in human nostrils among different people. Ultimately, we choose the nostril as the feature point to accomplish our approach [16]. The reason we choose nostril is the most stable and easily recognizable feature point in human face.
There are a lot of approaches to locate human face in an image, such as [17]. So we do not discuss the approach in which we get the face region in an image (the blue box in Figure 2). According to previous knowledge, we divided a region from the blue box in Figure 2 which is from the width of face region 2/7 to 5/7, height of face region 1/2 to 3/4 (the red box in Figure 2). This region must contain human nose and nostrils.

In the prior chapter, we have defined a "feature point" must be a window which has a corner. So we can scan all windows in the ROI image, calculated the two eigenvalues of $G (\lambda_1$ and $\lambda_2$), and selected several best windows. In a nose ROI, we can select several "feature points". Some of them are nostrils, and others are not. For a example, in Figure 2, we selected the best 4 "feature points" (green points). Two of them are nostrils.

At the beginning of our approach, we trained a classifier to identify nostril in nose ROI. In our approach, we only locate and track ONE nostril, there is no difference between the left nostril and the right nostril. So our nostril classifier does not distinguish whether a feature point is the left or right nostril, we just identify a point is a nostril or not.

In the nostril classifier training phase, GentleBoost feature templates are learned through the use of a representative set of positive (nostril points) and negative (non-nostril points) examples. As positive or negative examples for the nostril feature point, we used $5 \times 5 = 25$ image patches centered on each feature point. Finally, we have 100 positive and 200 negative point examples, meaning that we had a train set included $5 \times 100$ positive examples vectors and $5 \times 200$ negative examples vectors. That is a $1500 \times 8281 (13 \times 13 \times 8 \times 6 + 13 \times 13)$ size matrix. Now we have the classifier which can locate nostril quickly (almost in several millisecond per frame) and actually (almost a recognition rate of 100%).

Figure 1. Images of face and nostril
In a video, we select best 4 "feature points" in the first frame. Then, use the nostril classifier, we can identify whether a point is nostril or not in the selected 4 points. If none of the 4 points is identified as a nostril, we do the approach again in next frame.

We can extract a nostril's coordinates in frames by tracking the nostril point through the use of LK algorithm. Assuming in a frame, the coordinates of nostril is $I_t(x, y)$, in the next frame, the coordinates will change into $I_{t+1}(x+\Delta x, y+\Delta y)$. We extract the value of the change of coordinate as:

$$ S_{t+1} = (\Delta x, \Delta y), $$

which can directly reflects the movements of the nostril from one frame to the next.

We defined 6 types of head movements: nodding, shaking, bowing, turn to the left, turn to the right and remaining still. Different from facial expressions, a head movement is a continuous action, which means we cannot identify the movement in one single frame or image. So we selected 10 consecutive frames and extract every corresponding to each frame. That is a vector include 20 features, we named it VOCD (Vector of Coordinate Difference). We can extract VOCDs from videos, such vectors can represent the kind of head movement which it extracts.

In the movement classifier training phase (Figure 3. (b)), We can extract 6 groups of VOCD to train 6 GentleBoost classifiers to identify nod, shake, bow, turn face left, turn face right and non-movement (Figure 3. (c)). We named the classifiers as Clsnod, Clsshake, Clsbow, Clsleft, Clsright, Clsnormal. VOCDs in each group that have typical head movements are extracted from the videos manually which have the typical head movement correspondingly. To train a classifier, for a example, we can use VOCDs in the group corresponding nod movement as positive examples and VOCDs in the other five groups as negative examples. In this way, we trained 6 classifiers to identify head movement.

Eventually, in the final testing phase, we located and tracked ONE nostril at first. Then, we extract VOCDs from videos. Because VOCDs were extracted from 10 frames (Figure 3. (d)), so the 1st VOCD was extracted at the 10th frame (1st frame to 10th frame). We predicted that VOCDs used the 6 classifiers we had trained (Figure 3. (e)). After predicting the VOCD, we identified the movement as the classifier which had the highest response (with the largest positive sum $F'(x_i)$ in (16)). If no classifier had a positive response, we also identify the movement as "NORMAL" (non-movement).
(a) We get nose ROI in a frame of a video and track nostril in the video. (b) We extract VOCDs from six types of videos. (c) We train 6 classifiers use the VOCDs. (d) In testing phase, we extract VOCDs from test videos. (e) We classify VOCDs by using the classifiers we had trained, then output the identification results.

Figure 3. Outline of our approach.

4. Experiments and Results

4.1. Training Set

Our method was obtained through a database of videos captured in our laboratory. We captured 100 videos from over 20 subjects, male and female, at ages from 25 to 60 years old, and have different facial expressions. In each video, a person was asked to do the action twice: nodding--facing the front--shaking--facing the front--bowing--facing the front--turning the face to the left--holding--facing the front--to the right--holding--facing the front. For each group of VOCDs, we extracted 500 VOCDs from 50 different videos. That is, to each classifier, there was a training set include 500 positive examples and 2500 negative examples.

All videos we captured are 640×480, and the width of a face in frame are in the range of 180 pixels to 250 pixels. Thus, in the case that the width of a face is larger or smaller than it in the database, we cannot guarantee that the method's performance reported below will remain the same. Maybe we can resize the face into the same size and convert VOCD with the same proportion. The actual influence of such occurrences on the performance of the method is, however, the matter of future experimental studies.

4.2. Experimental Results

To evaluate the performance of the method, we tested the classifiers in two phases. In the first stage, we divided the 100 videos mentioned previously into two groups randomly, each group had 50 segments. We named them as test set 1 and test set 2. We used set 1 as the training set and set 2 as the testing set, and exchanged them as testing set and training set to test our approach again. In the second
stage, we used the 100 videos as training set, and captured another 50 videos from different 10 persons in test set 1 & 2 as testing set named final test set. In final test set, we do not asked a person to do an ordered series of actions, but to do any 10 actions of the head movement. After captured, the person was asked to declare what movements he had acted. If a movement is identified correctly in one frame in the series of frames of movement, we regarded the identification as a "SUCCESS". If no frame was identified as any movement in the series frames, or a frame or more were not identified correctly, we regarded the identification as an "ERROR".

The first stage was repeated 5 times. Each time we selected test set 1 and test set 2 randomly to train the classifiers. The results of the 5 times have not any difference. The identification rates of stage 1 are shown in Table 1, and the rates of stage 2 are shown in Table 2. In the tables, TL means turn face left, TR means turn face right, NI means not identified, NM means non-movement.

<table>
<thead>
<tr>
<th>Table 1. Identification rates of stage 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nod</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Nod</td>
</tr>
<tr>
<td>Shake</td>
</tr>
<tr>
<td>Bow</td>
</tr>
<tr>
<td>TL</td>
</tr>
<tr>
<td>TR</td>
</tr>
<tr>
<td>NM</td>
</tr>
</tbody>
</table>

Average rate for all movement is 94.3%

<table>
<thead>
<tr>
<th>Table 2. Identification rates of stage 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nod</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Nod</td>
</tr>
<tr>
<td>Shake</td>
</tr>
<tr>
<td>Bow</td>
</tr>
<tr>
<td>TL</td>
</tr>
<tr>
<td>TR</td>
</tr>
<tr>
<td>NM</td>
</tr>
</tbody>
</table>

Average rate for all movement is 91.4%

From the data in the table we can see that the identification rates of our proposed method to identify nodding and shaking is higher than bowing and face-turning. It can be improved through adjusting the length of VOCD. There were 3 face-turning actions in Table 1 that were not identified. After analysis the video we found that the reason was that the speed of the 3 actions were so slow that the movements were identified as non-movement.

5. Conclusion

In this paper we present a robust, highly accurate method for head movement recognition based on LK algorithm and GentleBoost. By positioning and tracking the nostril, our method can identify not only nodding and shaking but also bowing and face-turning. When tested on videos captured from different person, the method has achieved average recognition rates not less than 91.4%.

In future work we will investigate try to identify the movement for different sizes of faces. Also, we plan to conduct extensive experimental studies using other publicly available face databases.

6. References