A GP-ELM Gesture Recognition Algorithm for Tactile Animation

Yali Ren, Huiling Chen, Liguo Shuai
School of Mechanical Engineering, Southeast University, Nanjing 211189, PR China
Corresponding author: Yali Ren. Email: 190782384@qq.com. Tel: +86 13776683628.

Abstract: Artificial Neural Network (ANN) is one of the most important techniques in gesture recognition with high accuracy, but its slow response limited its real-time applications. Extreme Learning Machine (ELM), a novel ANN with fast learning ability, has the potential to solve the above problem. The extant recognition methods based on ELM focus on Local Variant Features (LVF) of gesture images and a small number of data are used to represent many features. However, the locality of these features may lead to an inadequate representation of real world information. In this paper, a new algorithm based on Global Pixels (GP) and ELM is proposed. It includes two parts: image pre-processing and GP-ELM identification. Through the processing of segmentation, filtering and standardization, a raw gesture image is transformed into a binarization and unified-size image. And then the GP of the picture are used as the ELM inputs to be trained or identified. The algorithm using GP instead of LVF for training simplifies the pre-processing without losing information. Experiment results show that the average recognition accuracy can reach 96.67%, while the recognition time is only 0.041 milliseconds. With more hidden nodes, the accuracy can be improved to be 100% with negligible increase in recognition time.

1. Introduction
The tactile animation is a fusion of computer science, animation arts, and tactile existence. On the technology side, the tactile animation device (TAD) for animation show comprises several parts: a camera for gesture-collection, a tactile glove for animation-play, a computer-controller unit as the smart center for data processing and system control and so on. When the operator touch the virtual object, the device will feed back real-time data to the controller. But a big time delay in gesture recognition will cause the system unstable. So how to develop an effective gesture recognition algorithm is a current key problem in this field.

The approaches for gesture recognition include template matching, Hidden Markov Models (HMMs) and Artificial Neural Networks (ANN). Nguyen-Dinh et al. [1], in 2012, proposed two recognition algorithms with template matching methods, which utilized the Longest Common Subsequence (LCSS) to improve robustness against noise for online gesture recognition. However it has bottleneck in improving recognition accuracy because it lacks the ability of enumerating all the features. HMM is a statistical algorithm widely used in voice recognition, voice-character conversion, natural language processing etc. But HMM only adopts contour features of the gesture picture, which limits the amount of free parameters to be estimated. As to ANN, In 2013, Toronow et al. [2] developed an interactive multi-agent robot system based on gesture recognition. However ANN takes a long time in model training and data analyzing, which may cause a remarkable time delay and make the system...
unstable. In 2004, a novel single hidden neural network named ELM was presented by Huang \cite{3}. It was able to compute efficiently and converge globally \cite{4}.

In this study, with higher accuracy and lower time consumption in hand gesture recognition, a GP-ELM algorithm is proposed. It employs the global pixels (GP) as gesture data and adopts ELM as identification approach.

2. Methodology

The process of gesture identification based on GP-ELM algorithm mainly consists of two parts: image preprocessing and GP-ELM recognition. In image preprocessing, there are three sequential steps: segmentation, filtering and standardization. The original gesture image captured by the camera is $320 \times 240$ pixels. After segmentation and filtering, it will be trimmed and finally resized to a $20 \times 20$ standard image with the most meaningful pixels included. For GP-ELM recognition, the input data for the model are the global pixels which are converted into a one dimensional array. There will be 400 nodes if all pixels are treated as the input, which is acceptable for ELM. The output of ELM is a one dimensional array with 7 elements representing seven different hand gestures. After being trained sufficiently, the ELM model will be well adjusted and ready for identification.

In this paper, model training utilized lots of pictures, including seven typical hand gestures of FIVE, as shown in Figure 1. These gestures are encoded by numbers from 1 to 7. The ideal output array for each gesture can be defined by a one-dimensional array $t_i$ with seven elements as:

$$t_i = (t_{i1}, t_{i2}, t_{i3}, t_{i4}, t_{i5}, t_{i6}, t_{i7})^T$$

And each element of this array can only be 0.1 or 0.9 in value.

![Figure 1. Gestures: FIVE, STONE, GOOD, ONE, OK, VICTORY, and GUN](image)

In this paper we adopted MATLAB simulations to investigate the relationship between the recognition error and the number of hidden nodes and find a reasonable range of the number of hidden nodes in ELM.

2.1 Image preprocessing

2.1.1 Segmentation and filtering

The first step of the algorithm is segmentation, which is to separate the raw gesture picture into the foreground and the background. Now skin color segmentation algorithm has become a significant method in gesture recognition.

From Hsu’s study \cite{5}, comparing the skin color distribution in color spaces, RGB, HSV, and YCbCr, the effect of the clustering performance in YCbCr space was found to be more productive. Wilson \cite{6} indicated that the Gaussian density function was of favorable approximation ability on the condition of a small number of samples. So the single-peak Gaussian model (SGM) is adopted during the segmentation in this paper.

A Gaussian joint probability density function in YCbCr space can be defined as

$$p(x | \text{skin}) = \frac{1}{2\pi |C|^2} \exp\left[-\frac{1}{2}(x - u)^T C^{-1} (x - u)\right]$$

Where $x$ is the chromaticity vector of each pixel; $u$ is the mean statistical vector, which has two elements: $C_r$ and $C_b$; and $C$ is the statistical covariance matrix.

$$C_r = \frac{1}{N} \sum_{i=1}^{N} C_{r_i}, C_b = \frac{1}{N} \sum_{i=1}^{N} C_{b_i}, u = (C_r, C_b)^T$$

\[ (3) \]
According to the statistical data of 20 hand skin pictures collected from participants, we get: \( u = (153.3975, 105.2072)^T \) and \( C = E[(x - u)(x - u)^T] = \begin{bmatrix} 21.616 & -11.5271 \\ -11.5271 & 12.415 \end{bmatrix} \).

With equation (2), \( p(x | \text{skin}) \) is worked out, and it is generally more than 0.95. The Gaussian distribution model (as shown in Figure 2), shows that the values of \( C_b \) and \( C_r \) of the skin color are in a very small range, which suggests a good clustering performance.

In practice, the reasonable threshold value \( p_t \) should be determined in advance. Once a \( p(x | \text{skin}) \) is bigger than \( p_t \), the corresponding pixel should be the pixel inside the hand skin, otherwise it might be noise or a pixel in the background. If \( p_t \) is too small the segmentation may leave many noises and cause difficulty in noise reduction and stray pixel-blocks removal. On the other hand, if it is too big, a great deal of meaningful pixels might be discarded, and the hand gesture image might be destroyed. In tactile animation, \( p_t \) may be set at 0.8 if the background of hand gesture is in good condition.

\[ C = E[(x - u)(x - u)^T] = \begin{bmatrix} \sigma_c^2 & \sigma_{cC} \\ \sigma_{Cc} & \sigma_C^2 \end{bmatrix} \]

(4)

2.1.2 Filtering

The picture after segmentation is quite clear in shape with some granular noise caused by illumination difference, light distribution and light reflection along the edge. The noise can be reduced effectively by filtering with morphological method.\(^7\) A structural unit of 3×3 is employed in image corrosion and expansion. Figure 4 shows the result of the final gesture image of segmentation.

After morphological process, there might be one or more stray pixel-blocks in white in the picture besides the biggest pixel block of hand image (as shown in Figure 5). This may cause a failure in the subsequent gesture recognition. As the hand image is the only biggest pixel-block in the picture, we may separate pixel-blocks from each other by pixel-relationship, and sort all pixel-blocks in area by
calculating the pixel-amount of each pixel-block. Then, we may remove the stray pixel-blocks by converting them into black except for the biggest pixel-block.

In Figure 5, the top left endpoint of the hand gesture image is treated as the origin point O, and the top edge is defined as X axis while the left edge is defined as Y axis. There are two pixel values in the picture, one is 0 representing white and another is 255 representing black. In order to find out the stray pixel-blocks, a two dimensional array $PB_{320 \times 240}$ with the same size as the gesture image is constructed, where the pixel value 255 is converted to ‘1’, while 0 remains unchanged.

The classification of the stray pixel-blocks is checked by the element values in $PB$ one by one in the strict order of position, from up to down and from left to right. When the first ‘1’ is found, we keep the ‘1’ unchanged to indicate that the first pixel-block is found, and then save all elements in this row in $CB$ as a backup for later analysis. The other elements in this row should be checked sequentially. When a new ‘1’ is found, we should judge whether it is adjacent to the previous ‘1’. If YES, the value ‘1’ keeps unchanged to show that it also belongs to pixel-block 1st. Otherwise the value ‘1’ is changed to ‘2’ to indicate that a new separated pixel-block is found, and it belongs to pixel-block 2nd, and the so on, until all elements have been examined.

When all elements of $PB$ are checked over, the array is changed as equation (5).

$$PB = \begin{bmatrix}
... & 0 & 1 & 1 & 1 & ... & 0 & 2 & ... & 0 \\
... & 0 & 1 & 1 & 0 & ... & 2 & 2 & ... & 0 \\
... & 1 & 1 & 0 & 3 & ... & 2 & 2 & ... & 0 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & 0
\end{bmatrix}_{320 \times 240} \quad (5)$$

Then the frequency of each serial number will be got, and the results will tell the coverage of each pixel-block. The biggest frequency value can be obtained by sorting these frequency values. It shows how many pixels the hand gesture image occupies in the picture. The total number of pixels in the biggest stray pixel-block can also be obtained.

In general, the coverage of the hand gesture image should be at least twenty times larger than the biggest stray pixel-block, so that we may change all stray pixel-blocks into ‘0’ in $PB$ according to the above results while keeping the hand gesture block unchanged if the times rate is met. Otherwise, the picture should be discarded because there might be something wrong with the camera, and a detailed check may be needed.

2.1.3 Standardization

After the matrix $PB$ standing for the clean hand gesture picture is obtained, we should trim the unmeaningful black margin by identifying the edge of the hand gesture pixel-block, that is, the top-line, the bottom-line, the left-column and the right-column. The top-line is the first non-zero row array in $PB$, and the bottom-line is the last non-zero row array. The left-line is the first non-zero column array, and the right-line is the last non-zero column array. According to the above four edge lines, the $PB$ standing for the clean hand gesture picture can be trimmed to the unique compact matrix $CM$. However, it is very large in size and may vary for different hand gesture images. As the standard matrix $SM$ for GP-ELM is $20 \times 20$, we should perform standardization (Figure 6) in the following steps.

![Figure 5. Gesture image with stray pixel-blocks](image1)

![Figure 6. Standardization](image2)
1. Ratio calculation

Provided that the size of $CM$ is $M_1 \times N_1$, where $M_1$ is the number of the rows, and $N_1$ is the number of columns of $CM$, the ratios for size regulation in row and column are $R_{\text{row}} = M_1 / 20$ and $R_{\text{column}} = N_1 / 20$. Where $R_{\text{row}}$ is the ratio for row regulation, and $R_{\text{column}}$ is the ratio for column regulation.

2. Row pixel regulation

Each pixel in the $20 \times 20$ matrix is a mapping of the corresponding pixels in $CM$, and its value is an average of many row pixels, which can be computed by the following equation,

$$\text{pixel}_{ij} = \frac{1}{\sum_{k=1}^{R_{\text{column}} \times (j-1) + 1.5}} \sum_{k=1}^{R_{\text{column}} \times j + 0.5} \text{pixel}_k$$

(1 ≤ $i$ ≤ $M_1$, 1 ≤ $j$ ≤ 20) (6)

3. Column pixel regulation

The column pixel regulation is similar to the row pixel regulation, and it can also be computed by the equation as follows,

$$\text{pixel}_{ij} = \frac{1}{\sum_{k=1}^{R_{\text{column}} \times (j-1) + 1.5}} \sum_{k=1}^{R_{\text{column}} \times j + 0.5} \text{pixel}_k$$

(1 ≤ $i$ ≤ 20, 1 ≤ $j$ ≤ 20) (7)

As $\text{pixel}_{ij}$ is decimal, and there is a need of round-off with the following equation,

$$\text{pixel}_{ij} = \lfloor \text{pixel}_{ij} + 0.5 \rfloor$$ (8)

When the final $20 \times 20$ matrix $SM$ is obtained, the global pixels can be converted into a one dimensional array as equation (9), and each element can be used as the input node of GP-ELM in sequence.

$$P = (x_1, x_2, ..., x_j, ..., x_{400})^T = (\text{pixel}_{11}, \text{pixel}_{12}, ..., \text{pixel}_{120}, \text{pixel}_{21}, ..., \text{pixel}_{200})^T$$ (9)

2.2 GP-ELM recognition

Figure 7 is the topology of GP-ELM algorithm. There are 400 input nodes as data-in and 7 output nodes as data-out.

![Figure 7. The topology of GP-ELM](image-url)

For N arbitrary distinct samples $(x_j, t_j)$, where $x_j = [x_{j1}, x_{j2}, ..., x_{j400}]^T \in \mathbb{R}^{400}$ is the input data array with 400 elements, $t_j = [t_{j1}, t_{j2}, ..., t_{j7}]^T \in \mathbb{R}^7$ is the output data array with 7 elements, a standard ELM with L hidden nodes is mathematically modeled as (3).
\[ \sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i (w_i \cdot x_j + b_i) = o_j, \quad j = 1, \ldots, N \] (10)

where \( w_i = [w_{i1}, w_{i2}, \ldots, w_{i400}]^T \) is the weight vector connecting the \( i \)th hidden node and the input nodes, \( \beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{i7}]^T \) is the weight vector connecting the \( i \)th hidden node and the output nodes, and \( b_i \) is the threshold of the \( i \)th hidden node, and \( g(x) \) means the activation function, \( o_j = [o_{j1}, o_{j2}, \ldots, o_{j7}]^T \) is the model output of ELM.

If the GP-ELM satisfies the \( N \) samples exactly, the error expectation should be zero, that is,
\[ E = \sum_{j=1}^{N} |o_j - t_j | = \sum_{j=1}^{N} \sum_{m=1}^{7} (o_{jm} - t_{jm})^2 = 0 \] (11)

Then,
\[ E = \sum_{j=1}^{N} \sum_{m=1}^{7} \left( \sum_{i=1}^{L} \beta_{im} g(w_i \cdot x_j + b_i) - t_{jm} \right)^2 \] (12)

Here, \( g(x) = 1/(1 + e^{-x}) \) is infinitely differentiable, and \( w_i, b_i \) are decimal numbers assigned between 0 and 1 randomly, then we get \( \beta = H^T T. \) Where \( H^T \) is the generalized inverse of \( H, \) and \( H \) is the hidden layer output matrix,
\[ H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix} \quad T = \begin{bmatrix} t_{11} & t_{21} & \cdots & t_{N1} \\ t_{12} & t_{22} & \cdots & t_{N2} \\ \vdots & \ddots & \vdots \\ t_{17} & t_{27} & \cdots & t_{N7} \end{bmatrix} \] (13)

As the number \( L \) of hidden nodes has an effect on matrix \( H, \) and it may influence the output error indirectly, there is a recommendation value for the number of hidden nodes \( L. \) In practice, the best way to determine the reasonable number of hidden nodes is by experiments, as detailed in the next section.

3. Experiments

3.1 Preparation

3.1.1 Platform

The diagram of the tactile animation platform [8] is shown in Figure 8. It consists of five parts, that is, a camera for hand picture collection, a glove for tactile vibro-stimulation on fingers, a smart center for data processing and system control, and a computer for program uploading and an interaction area for the display of hand gestures.

![Figure 8. Tactile animation platform](image)
In addition, there are 5 round vibrators planted inside each finger cap of the glove. They are located at the finger-tip side. The alignment of vibro-actors should not interfere with fingers’ movement, and allow fingers to stretch in a wide range of angles.

3.1.2 Participants and samples
10 participants with normal vision, healthy arms and some knowledge of basic communication gestures are recruited for gesture collection. Each of them is asked to make 20 available examples for each kind of gesture.

Then 200 available images are required for each gesture, and 100 of them randomly chosen are used for training and 50 of the rest are employed for verifying. Correspondingly, a gesture database is built with 1050 (150 × 7 gestures = 1050) examples including 700 training images and 350 testing images.

3.2 The number of hidden nodes
The figure 9 shows the error curves for different ranges of $L$ with 100 training samples and 50 verifying samples. The curves are derived by simulation with ELM algorithm implemented in MATLAB. Figure 9 shows that when the number of hidden nodes is below 700, there are many unstable crests on the curves, and local or global peak values appear just near the point of 700. After the peaks the errors are monotonously decreasing with the increase of $L$ with a small and stable allowance. So the gesture recognition system should work better when $L$ is above 700, and we chose to set $L = 750$ in the following experiments.

3.3 Experiments
Several experiments have been conducted with the tactile animation platform to investigate the recognition accuracy ($RA$), the training time of 700 samples ($TT$), and the mean recognition time ($\overline{RT}$) of GP-ELM algorithm for tactile animation. The computer is configured with Intel® Core™ i7-4790K CPU @ 4.00GHz; RAM 16.0GB, and 64 bit Windows 7 Operation System.

As mentioned above, $L$ is set as 750 in the following experiments.

$RA$ is the percentage of correct instances vs. total instances during gesture recognition experiments,

$$RA = \frac{\text{correct instances}}{\text{total instances}} \times 100\%$$

$\overline{RT}$ is the average value of $RT_i$, which can be computed by the sum of each verifying sample’s recognition time and then divided by the number of total testing samples, that is 30,

$$\overline{RT} = \frac{1}{30} \sum_{i=1}^{30} RT_i$$
Where $RT_i$ is the recognition time for the $i_{th}$ hand gesture.

The GP-ELM is firstly trained with $100\times7$ samples, and then $30\times7$ samples are used for verification. During the verification, each sample will produce an output matrix $\mathbf{o}_i = (o_{i1}, o_{i2}, o_{i3}, o_{i4}, o_{i5}, o_{i6}, o_{i7})^T$. With the equation (12), $E_q (q=1-7)$ can be obtained, and then $E_{\min}$ can be worked out by,

$$E_{\min} = \min\{E_q, q=1-7\} = \min\{E_1, E_2, E_3, E_4, E_5, E_6, E_7\}$$

(16)

Where $q$ is the sequence number of hand gesture from FIVE to GUN, and $E_q$ is the error of the $q_{th}$ gesture. Table 1 is the list of RA, $TT$, and $\overline{RT}$ for different gestures.

The table 1 shows that the GP-ELM algorithm has favorable accuracy in typical gesture recognition, and can reach 96.67% on average, while the training time is only 2.4 seconds, and the average recognition time for each gesture is only 0.041 milliseconds. Among them, FIVE, STONE, GOOD and ONE can be classified easily with a correct ratio of 100%, and followed by VICTORY with a correct ratio of 96.67%, while OK is relatively harder to recognize with a correct ratio of 86.67%. However this result is still encouraging. The experiments show that OK may most likely be confused with GUN and VICTORY. This is a potential area for improvement in the future.

Table 1. RA, $TT$, and $\overline{RT}$ for different gestures (with 700 hidden nodes)

| Gestures | Total instances | Correct instances | RA | $TT$ | $\overline{RT}$ |
|----------|----------------|------------------|----|------|----------------|
| FIVE     | 30             | 30               | 100% |      |                |
| STONE    | 30             | 30               | 100% |      |                |
| GOOD     | 30             | 30               | 100% |      |                |
| ONE      | 30             | 30               | 100% | 2.4s | 0.41*10^-4s    |
| OK       | 30             | 26               | 86.67% |      |                |
| VICTORY  | 30             | 29               | 96.67% |      |                |
| GUN      | 30             | 28               | 93.3% |      |                |
| Total    | 210            | 203              | 96.67% |      |                |

Further experiments with 1,000 hidden nodes are also conducted to investigate the influence of a bigger $L$, and the results (as shown in Table 2) are fairly good with a correction of 100% for each gesture, while the response time of $TT$ and $\overline{RT}$ has a monotonous increase with $L$. However the increase is almost negligible: 0.052 milliseconds. So a big $L$ may be recommended in gesture recognition.

Table 2. RA, $TT$, and $\overline{RT}$ for different gestures (with 1,000 hidden nodes)

| Gestures | Total instances | Correct instances | RA | $TT$ | $\overline{RT}$ |
|----------|----------------|------------------|----|------|----------------|
| FIVE     | 30             | 30               | 100% |      |                |
| STONE    | 30             | 30               | 100% |      |                |
| GOOD     | 30             | 30               | 100% |      |                |
| ONE      | 30             | 30               | 100% | 2.4s | 0.52*10^-4s    |
| OK       | 30             | 30               | 100% |      |                |
| VICTORY  | 30             | 30               | 100% |      |                |
| GUN      | 30             | 30               | 100% |      |                |
| Total    | 210            | 210              | 100% |      |                |

4. Conclusion
This paper presented a new GP-ELM algorithm to tackle the hand gesture recognition problem for tactile animation applications. The algorithm was the first attempt of treating the global pixels as input nodes of ELM. Experiments show that GP-ELM has a good performance in typical hand gesture
recognition with a correct ratio of 96.67% on the condition of 400 input nodes, 750 hidden nodes, and 7 output nodes. And a higher correction at 100% can be obtained if the number of hidden nodes $L$ is increased to be over 1,000 with negligible increase in response time. However, this result is obtained strictly under the condition of typical hand gestures. Future work will be pursued to make it applicable in other cases such as body recognition and face recognition. Another potential area is to investigate how to keep almost the same performance on the condition of bigger PB matrix in hand gesture recognition.

Acknowledgements
This work is supported by the National Natural Science Foundation of China (61175069), the National Science Foundation's Course, Curriculum, and Laboratory Improvement (CCLI) program (0737173), the Natural Science Foundation of Jiangsu Province of China (BK20131205) and the Prospective Project of Jiangsu Province for Joint Research (SBY201320601).

References
[1] Long-Van Nguyen-Dinh, Roggen D, Calatroni A, Troster G 2012, Improving online gesture recognition with template matching methods in accelerometer data. International Conference on Intelligent Systems Design and Applications, ISDA (12) 831-836
[2] Tornow M, Al-Hamadi A, Borrmann V 2013, A multi-agent mobile robot system with environment perception and HMI capabilities. IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2013 252-257
[3] Huang GB, Zhu QY, Siew CK 2006, Extreme Learning Machine: Theory and Applications. Neurocomputing 70 489-501
[4] M. Kiran Kranthi and T. Shyam Vamsi 2014, Hand Gesture Detection and Recognition Using Affine-Shift, Bag-of-Features and Extreme Learning Machine Techniques. Advances in Intelligent System and Computing 247 181-187
[5] Hsu RL, Jain AK 2002, Face Detection in Color Images. IEEE Transactions Pattern Analysis and Machine Intelligence 24(5) 696-706
[6] Wilson R 1999, Multi-resolution Gaussian Mixture Models: Theory and Application. Research Report RR404, Department of Computer Science, University of Warwick, UK 1-10
[7] Tang CQ, Lv HB, et al 1990, Mathematical morphology methods and applications. Science Press 1 76-90
[8] Shuai LG, Gu CX, Chen HL, Zhou ZT 2009, Preliminary research on tactile cartoon animation and its showing art. Proceedings of 2009 IEEE 10th International Conference on Computer-Aided Industrial Design and Conceptual Design, E-Business, CAID and CD 2009 1896-1899