

Gesture recognition through angle space

F. DADGOSTAR, A. SARRAFZADEH

*Institute of Information & Mathematical Sciences
Massey University at Albany, Auckland, New Zealand[†]*

As the notion of ubiquitous computing becomes a reality, the keyboard and mouse paradigm become less satisfactory as an input modality. The ability to interpret gestures can open another dimension in the user interface technology. In this paper, we present a novel approach for dynamic hand gesture modeling using neural networks. The results show high accuracy in detecting single and multiple gestures, which makes this a promising approach for gesture recognition from continuous input with undetermined boundaries. This method is independent of the input device and can be applied as a general back-end processor for gesture recognition systems.

1 Introduction

Gesture recognition systems identify human gestures and the information they convey. Although relying on gesture as the primary source of command input to computers may sound like science fiction, the technology has rapidly progressed in some areas such as virtual reality, by relying on special hardware and wearable devices. This hardware is often not cost-effective and is infeasible for some applications; consequently, gesture recognition based on alternative methods of data acquisition is being considered. In this article we introduce a novel method for gesture recognition in 2D space which we used for interpreting hand and head movements as gesture commands to a vision-based user interface. In Section 2, we describe the research background and present a brief literature review. In Section 3, we discuss our approach for gesture modeling, and in Section 4, we present our validation approach and the results of our experiments. Section 5 presents a summary and our conclusions.

2 Research background

The gesture recognition problem consists of pattern representation and recognition. The hidden Markov model (HMM) is used widely in speech recognition, and a number of researchers have applied HMM to temporal gesture recognition. Yang and Xu (1994) proposed gesture-based interaction using a multi-dimensional HMM. They used a Fast

[†] Optional Email addresses: f.dadgostar@massey.ac.nz; h.a.sarrafzadeh@massey.ac.nz

Fourier Transform (FFT) to convert input gestures to a sequence of symbols to train the HMM. They reported 99.78% accuracy for detecting 9 gestures.

Watnabe and Yachida (1998) proposed a method of gesture recognition from image sequences. The input image is segmented using maskable templates and then the gesture space is constituted by Karhunen-Loeve (KL) expansion using the segment. They applied Eigen vector-based matching for gesture detection.

Oka, Satio and Kioke (2002) developed a gesture recognition based on measured finger trajectories for an augmented desk interface system. They used a Kalman-Filter for predicting the location of multiple fingertips and HMM for gesture detection. They have reported average accuracy of 99.2% for single finger gestures produced by one person.

Ogawara et al. (2001) proposed a method of constructing a human task model by attention point (AP) analysis. Their target application was gesture recognition for human-robot interaction.

New et al. (2003) proposed a gesture recognition system for hand tracking and detecting the number of fingers being held up to control an external device, based on hand-shape template matching. Perrin et al. (2004) described a finger tracking gesture recognition system based on laser tracking mechanism which can be used in hand-held devices.

They have used HMM for their gesture recognition system with an accuracy of 95% for 5 gesture symbols at a distance of 30cm to their device.

Lementec and Bajcsy (2004) proposed an arm gesture recognition algorithm from Euler angles acquired from multiple orientation sensors, for controlling unmanned aerial vehicles in presence of manned aircrew. Dias et al. (2004) described their vision-based open gesture recognition engine called OGRE, reporting detection and tracking of hand contours using template matching with accuracy of 80% to 90%.

Because of the difficulty of data collection for training an HMM for temporal gesture recognition, the vocabularies are very limited, and to reach to an acceptable accuracy, the process is excessively data and time intensive. Some researchers have suggested that a better approach is needed for use with more complex systems (Perrin et al., 2004).

3 Modeling the gesture in 2D space

Feature selection for pattern recognition requires considerations such as selecting the smallest number of features to adequately explain the pattern, and preferably invariant to some unfavourable factors.

McNeill (McNeill, 1992) defines a hand gesture as “the movement of a hand between two rests”. The hand movement itself carries a large amount of data, e.g. velocity, acceleration, direction and position. The larger the number of selected features, the more time required to process the data. This makes feature selection for gesture recognition an important, but challenging task.

3.1 Primary Assumptions and Requirements

In this research, we aim to detect single-hand gestures in two dimensional space, using a neural-network (NN). The selected features should be i) as small as possible in number, ii) invariant to input errors like vibrating hand, small rotation and scale which may vary from person to person or with different input devices. Here, our primary assumption is

that the position of the hand is tracked using an input device. The acceptable delay of the system is the end of each gesture, meaning that the pre-processing should be in real-time.

One of our goals in the design and development of this system is scalability in detecting a reasonable number of gestures and the ability to add new gestures in the future. To be able to do this analysis and feature selection from each individual gesture was not feasible. We preferred to take a general approach to feature selection from the input gestures. Considering the requirements discussed in this section we used sampling of the angle of the input gesture vector as explained in the following paragraphs.

3.2 Gesture Modelling

Each 2D gesture can be presented as a set of small connected movement vectors of the hand over time, such as $G = \{v_t | v_t = (A_t, A_{t+1}), A_t = (x_t, y_t)\}_{t=1..n}$. This representation makes the reconstruction of the input very accurate but not invariant to location. Our model for gesture representation is based on this method in polar coordinates, and using the angle of each vector which can be calculated using equation 1.

$$\theta = \text{ArcTan}((y_{t+1} - y_t) / (x_{t+1} - x_t)) \quad (1)$$

To quantize the value of θ , we used equal distances of 10° , thus each sample after quantization will have a value between 0 to 35. Therefore, different movement directions are represented by different sequences of integer values 0 to 35. The sequence data implicitly includes the time and the direction of the gesture movements. Figure 1a shows a simple hand movement. The density of the arrows in different parts of the movement presents the speed of the hand in those parts. Observe that the hand has had some vibrations in some parts, and the number of samples (arrows) in Figure 1a is considerably more than Figure 1b, which is the quantized version of the original movement. Using this technique, a gesture can be translated into a gesture signal (Figure 1c,d), which reduces the gesture recognition problem to a signal matching problem.

3.3 Robustness against slight rotation and relocation

Figure 1d, shows gesture data with slight rotation and relocation. The graph shows that the proposed modeling of the gesture is robust against relocation, because of the independency to the coordinates of the input. The interesting feature of this model the transformation of the rotation to a horizontal shift in the angle space.

3.4 Normalizing the Data

This modeling approach produces a variable number of samples for each individual user or sometimes for the same user, which is not ideal for classifiers. Most classifiers such as Neural-Networks (NN), Support Vector Machines (SVM), and Eigen Vector-based classifiers require a certain number of features to classify the pattern. Therefore, a normalization process to the input data is needed, to equalize the size of the input

dataset. The normalized data has two uses in this application, 1) for training a classifier, and 2) for feeding the classifier in the detection phase.

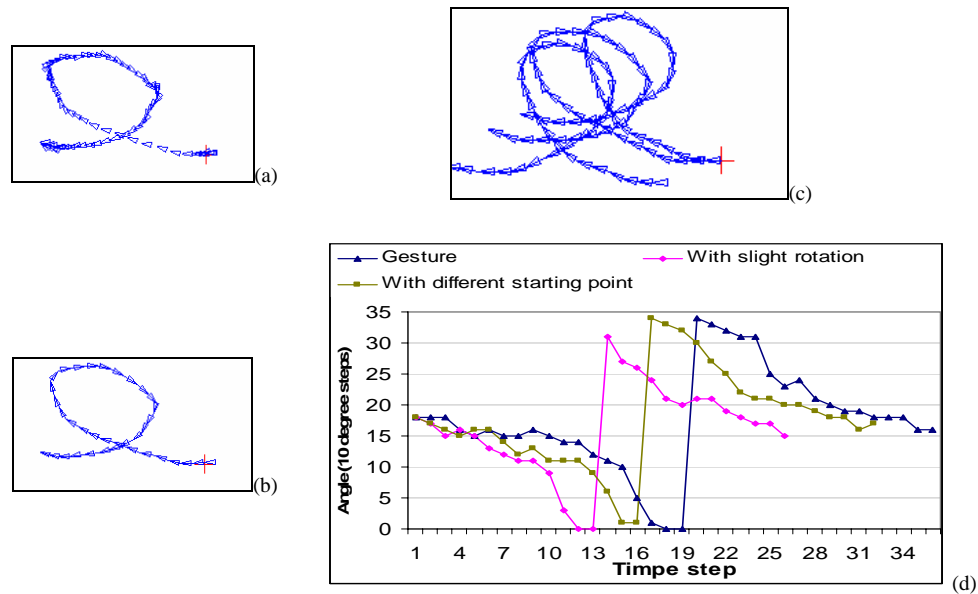


Figure 1. Gesture pattern, a) Original gesture move, b) reconstructed gesture using (c), c) collected data over time, d) gesture samples, e) gesture signals.

To normalize the data, about 200 samples for each gesture were manually prepared. In the first step, the average size of each gesture was considered as the normal size of the input data. Based on the calculated average size, an average vector was calculated and used as a fitness measurement for the rest of the training data. For signals with a size different from the average, a horizontal shift between -5 to 5 was considered such that the Euclidian distance of the sample signal and the average signal was made minimal. Then, by trimming and replicating the tail of the shifted signal, its size was adjusted to average size. The result of this step is a set of vectors with equal sizes suitable for training the classifier. In this research, we modeled two other gestures presented in Table 1. The normalization procedure was the same as for the first gesture.

We should note that another approach for signal matching is Dynamic Time Warping (Sakoe & Chiba, 1978) (DTW), which provides smaller Euclidean distance between the two signals, but requires the presence of the complete signal in advance. This requirement was less favorable, because of our consideration for dynamic feeding of the NN and automatic detection of the boundaries of the signal for further extensions of the application.

4 The Experiments - Gesture Detection

In the next step, the normalized data was used for training a NN. In this research we have tested the model for three gestures. For each gesture, we prepared a set of about 200 gesture signals as training data, and about 150 gesture signals as test data.

4.1 The First Experiment – Using Separate Classifiers for each Gesture

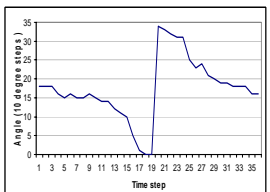
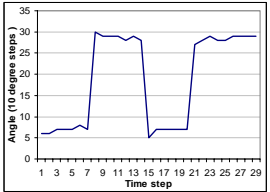
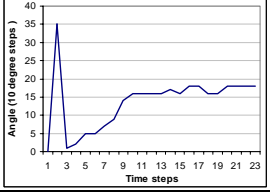
In the first experiment we used an unbiased multi-layer feed forward neural network with the following characteristics for each gesture:

- Number of inputs = size of the normal vector
- Two middle layers containing 10 and 3 neurons respectively
- One output

The input signals were normalized to [0 1] by dividing by 35. The weights of the neurons were randomly initialized. The output was translated to 0/1 for true/false values, using a HardLimit transform.

After training the NN, we applied the gesture signals of the test data to the NN. The results of this experiment are presented in Table 1.

Table 1. The results of the experiments

Gesture pattern	Gesture signal	Size of training dataset	Size of test dataset	Experiment 1		Experiment 2	
				Avg vector size	Accuracy	Avg vector size	Accuracy
(1)		225	126	31	100%	28	100%
(2)		221	113	29	100%	28	98.2%
(3)		254	96	23	99.6%	28	97.9%

4.2 The Second Experiment – Using a Single Classifier

The classification ability of separate NNs for gesture detection in the first experiment was promising. Therefore, in the next step, we organized another experiment to evaluate a single NN for classifying the same three gestures. The challenge of this experiment was the different sizes of the average vectors for each gesture. Instead of using the average size value for the normal vector, we used a constant size of 28 as the general vector size and then applied the normalization process to all the three training datasets. This means that for vectors smaller than this size, we had to add some automatically generated data that may cause decreased accuracy in the classification. Specifically, we expect a lower accuracy for gesture (3) that its average size is 20% smaller than the

proposed size. For this experiment, we used a multi-layer feed forward neural-network with the same characteristics as in experiment 1, as follows:

- 28 inputs
- Two middle layers containing 20 and 9 neurons respectively
- Three outputs

The outputs were translated to $[0, 1]$, using a HardLimit transform. The results of the second experiment are also presented in Table 1. The results presented here show the classification ability of the NN after 4000 iterations for training.

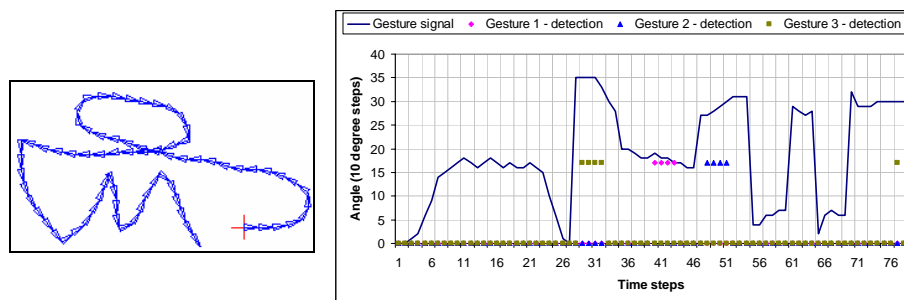


Figure 2. Continues gesture detection using NN

4.3 Gesture detection with continuous input

Although our original design was based on gesture detection with known average size, we also supplied the NN of experiment 2 with continuous input to observe its validity for gesture detection. With this approach the normalization process was not applied to the input and therefore less accuracy is expected. In the experiment, the input signals were used as the inputs of the NN. The inputs of the NN were initially set to 0, and each input angle was pushed to the input queue at the time of observation. Figure 2 shows the input gesture and the detection results. The sequence of the three gestures was correctly detected. However the boundaries of the gesture signal were not determinable because of the multiple detections of each gesture.

5 Summary and Discussion

In this paper we presented a novel approach for gesture detection. This approach has two main steps: i) gesture modelling, and ii) gesture detection. The gesture modelling technique which we presented here has some important features for gesture recognition including robustness against slight rotation, small number of required features, invariant to the start position and device independence. For gesture detection, we used a multi-layer feed-forward neural-network. The results of our first experiment show 99.72% average accuracy in single gesture detection. In the second experiment we used another NN for gesture classification, which shows 98.71% average accuracy for gesture detection. Based on the high accuracy of the gesture classification, the number of NN layers seems to be enough for detecting a limited number of gestures. However, more accurate judgment requires a larger number of gestures in the gesture-space to further validate this assertion. Our observation also shows that this technique is a potential

approach for continuous gesture classification. The gesture recognition technique introduced in this article can be used with a variety of front-end input systems such as vision-based input, hand and eye tracking, digital tablet, mouse, and digital glove.

References

- Dias, J. M. S., Nande, P., Barata, N., & Correia, A. (2004). *OGRE - open gestures recognition engine*. Paper presented at the Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on.
- Lementec, J.-C., & Bajcsy, P. (2004). *Recognition of arm gestures using multiple orientation sensors: gesture classification*. Paper presented at the Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on.
- McNeill, D. (1992). *Hand and Mind*: The University of Chicago Press.
- New, J. R., Hasanbelliu, E., & Aguilar, M. (2003). *Facilitating User Interaction with Complex Systems via Hand Gesture Recognition*. Paper presented at the Proceedings of the 2003 Southeastern ACM Conference, Savannah, GA.
- Ogawara, K., Iba, S., Tanuki, T., Kimura, H., & Ikeuchi, K. (2001). *Acquiring hand-action models by attention point analysis*. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA).
- Oka, K., Sato, Y., & Koike, H. (2002). Real-time fingertip tracking and gesture recognition. *IEEE Computer Graphics and Applications*, 22(6), 64-71.
- Perrin, S., Cassinelli, A., & Ishikawa, M. (2004). *Gesture recognition using laser-based tracking system*. Paper presented at the Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on.
- Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*, *IEEE Transactions on*, 26(1), 43-49.
- Watanabe, T., & Yachida, M. (1998). *Real time gesture recognition using eigenspace from multi-input image sequences*. Paper presented at the Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, Nara, Japan.
- Yang, J., & Xu, Y. (1994). *Gesture Interface: Modeling and Learning*. Paper presented at the IEEE International Conference on Robotics and Automation.

