1-1-2010

Dynamic hand gesture recognition system using moment invariants

Zhengmao Zou

University of Wollongong, zz942@uow.edu.au

Prashan Premaratne

University of Wollongong, prashan@uow.edu.au

Malin Premaratne

Ravi Monaragala

Nalin Bandara

Follow this and additional works at: <https://ro.uow.edu.au/engpapers>

Part of the [Engineering Commons](#)

<https://ro.uow.edu.au/engpapers/547>

Recommended Citation

Zou, Zhengmao; Premaratne, Prashan; Premaratne, Malin; Monaragala, Ravi; and Bandara, Nalin: Dynamic hand gesture recognition system using moment invariants 2010, 108-113.
<https://ro.uow.edu.au/engpapers/547>

Dynamic Hand Gesture Recognition System using Moment Invariants

Zhengmao Zou*, Prashan Premaratne*, Ravi Monaragala⁺, Nalin Bandara⁺, Malin Premaratne⁺⁺

*School of Electrical Computer and Telecommunications Engineering
University of Wollongong, North Wollongong, NSW 2522, Australia.

Email: prashan@uow.edu.au

⁺Centre for Research and Development, Army Cantonment, Panagoda, Sri Lanka.

⁺⁺Advanced Computing and Simulation Laboratory (AXL)
Department of Electrical and Computer Systems Engineering
Monash University, Clayton, Victoria, Australia.

Abstract— We have developed a dynamic hand gesture recognition system that can simplify the way humans interact with computers and many other non-critical consumer electronic equipment. The proposed system is based on the well-known “Wave Controller” technology developed at the University of Wollongong [1–3] and will revolutionize video gaming and consumer electronics control interfaces. Currently, computer interfacing mainly involves keyboard, mouse, joystick or gaming wheels and occasionally voice recognition for user input. These modes of interaction have restrained the artistic ability of many users, as they are required to respond to the computer through pressing buttons or moving other apparatus. Voice recognition is seen as unreliable and impractical in areas where more than one user is present. All these drawbacks can be tackled by using a reliable hand gesture recognition system that facilitates interaction between users and computers and other consumer electronic equipment in real time. This will further enhance the user experience as users no longer have any physical connection to the equipment being controlled. This system can also be extended to a sign language system for the benefit of the disabled including those with speech disabilities.

I. INTRODUCTION

With their ever-increasing processing power and plummeting costs, the use of computers and other microcontroller-based consumer electronics equipment in the average home is increasing. Whether in our lounge room, bedroom or office, a number of electronic devices may require commands to perform some valuable tasks. It could be the television set, the VCR or the set-top box waiting for our command to provide us with music or perhaps news and the command may reach them with a push of a button of a remote controller or a keyboard. People have long tried to replace such controllers with voice recognition or glove-based devices [4–8], with mixed results. Glove-based devices are tethered to the main processor with cables that restrict the user’s natural ability to communicate. Many of these approaches have been implemented to focus on a single aspect of gestures, such as hand tracking, hand posture estimation, or hand pose

classification, using uniquely coloured gloves or markers on hands or fingers [9–17]. Technological experts overwhelmingly agree that the bottleneck in computer and gaming adoption is the human-computer interface. This is evident from the popularity of the Nintendo gaming console, which has replaced its traditional game pad with a remote controller called “Wii”. This has increased computer usage and is increasingly engaging the older generation who do not like using the traditional keyboard and mouse to interact with gaming electronic entertainment units. This has further thrust human computer interaction (HCI) to a new level of sophistication and hand gesture recognition is seen at the forefront of that trend [2]. The development of the “Wave Controller” has demonstrated that traditional computer and consumer electronic controls such as remote controllers can be replaced effectively with a hand gesture recognition system.

Hand gesture recognition systems rely on computer vision to recognize hand gestures in the presence of any background clutter. These hand gestures must then be uniquely identified to issue control signals to the computer or another entertainment unit. During the past 18 years, many attempts have been made with mixed success to recognize hand gestures. Some of the drawbacks of the original systems were their inability to run in real time with modest computing power and low classification scores in gesture recognition. Most of the techniques relied on template matching [11] or shape descriptors and required more processing time than could be provided in real time. Furthermore, the users were restricted to wearing gloves or markers to increase reliability, and were also required to be at a preset distance from the camera. These restrictions meant that a practical system that could operate in a lounge room was unthinkable. In 2007, a truly practical system that was capable of running on modest computing power in real time was unveiled at the University of Wollongong and was hailed as a major breakthrough in hand gesture recognition system [2]. This research was distinguished from previous

attempts by a few marked differences:

1. using a minimum number of gestures offers higher accuracy with less confusion;
2. only low processing power is required to process the gestures, making the system useful for simple consumer control devices;
3. the system is very robust to lighting variations; and
4. the system operates in real time.

I. DYNAMIC GESTURE RECOGNITION

This research discussed here has significantly improved from our first attempt using static hand gesture system. The dynamic hand gesture recognition system proposed here is a real time system with moderate computing requirements while recognizing significantly more gestures than the seven known to the “Wave Controller” [1]. Dynamic gesture recognition also minimizes the problems associated with hand movements in static hand gesture recognition systems. The system is capable of rejecting unintentional gestures more easily than a static system because of the start and stop routines associated with dynamic gestures. A flow chart of the implementation steps of the system is depicted in Fig. 1.

As clearly outlined in the flowchart, when an image is acquired, it is processed and noise removed and classified immediately to see whether the captured gesture is ‘start’ so that successive gestures could be captured. Hence, any unintentional gestures will be disregarded by the system. If and when a ‘start’ gesture is recognized, the system will be ready to capture a second, third and many more to interpret them for further actions. The capturing and interpreting will cease once a ‘stop’ gesture is captured for one dynamic gesture and the system will again look for ‘start’ for further instructions. We have set the system to capture images every 250 milliseconds. In our new attempt to process and recognize images using C and C++ rather than MATLAB, we have succeeded in using less computing power and process more images in real time. We have used Microsoft Founding Classes (MFC) to implement the system Graphical User Interface (GUI) (Fig. 2) and OpenCV for image processing tasks. This has resulted in a ten fold speed up of the processing which facilitates dynamic gesture recognition. The following innovative steps highlight the uniqueness of the dynamic gesture recognition approach:

1. Hand gestures are performed against a non-uniform background; this will result in a very robust and practical system.
2. The distance between the camera and different gestures can vary, which will result in different-sized pictures; however, using moment invariants for feature selection will result in scale invariant features [1].
3. No markers or special gloves will be used.
4. Real-time implementation will use modest computing

power as all the processing is done using C and C++.

Fig. 1. Flowchart of Dynamic hand gesture system

Fig. 2. GUI of Dynamic gesture recognition

The image processing tasks remain essentially the same as in our first attempt developing a static gesture recognition system. This includes captured image is being preprocessed for noise removal using morphological filtering and skin region segmentation. The captured hand gestures from a real-time video stream must be processed before they can be interpreted by a computer. It is vital that the captured image is registered as a hand gesture using skin segmentation after removing the background of the image. The skin segmentation technique used in this research involves

converting the image from RGB format to YCbCr format [1]. Then a threshold filter is applied to remove ‘non-skin’ components. The major advantage of this approach is that the influence of luminosity can be removed during the conversion process, making the segmentation less dependent on the lightning conditions, which has always been a critical obstacle for image recognition. The threshold values will be obtained from the database. A number of sample points that represent skin patches and non-skin patches are then obtained. Gesture normalization uses the well-known morphological filtering technique using erosion combined with dilation [18]. The output of this stage is a smooth region of the hand figure, which is stored as a logical bitmap image.

Moment invariants are still retained as the major set of features and the number of gestures will be increased using dynamic gestures. It is envisaged that using dynamic gestures, the set of seven very accurate static gestures can be theoretically increased to 42, as one can perform two unique static gestures in a dynamic gesture. Two sets of dynamic gestures are shown in Figures 3 and 4.

II. FEATURE EXTRACTION OF THE HAND GESTURE

Image classification is a very mature field today. There are many approaches to finding matches between images or image segments. Starting from the basic correlation approach to the scale-space technique, they offer a variety of feature extraction methods with varying success. However, it is very critical in hand gesture recognition that the feature extraction is fast and captures the essence of a gesture in unique small data set. Neither the Fourier descriptor, which results in a large set of values for a given image, nor scale space succeed in this context. The proposed approach of

using moment invariants stems from our success in developing the “Wave Controller”. Gesture variations caused by rotation, scaling and translation can be circumvented by using a set of features, such as moment invariants, that are invariant to these operations.

The moment invariants algorithm has been recognized as one of the most effective methods to extract descriptive feature for object recognition applications and has been widely applied in classification of subjects such as aircrafts, ships, and ground targets, [19, 20]. Essentially, the algorithm derives a number of self-characteristic properties from a binary image of an object. These properties are invariant to rotation, scale and translation. Let $f(i, j)$ be a point of a digital image of size $M \times N$ ($i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$). The two dimensional moments and central moments of order $(p + q)$ of $f(i, j)$, are defined as:

$$m_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j)$$

$$U_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j)$$

Where

$$\bar{i} = \frac{m_{10}}{m_{00}} \quad \bar{j} = \frac{m_{01}}{m_{00}}$$

From the second order and third order moments, a set of seven (7) moment invariants are derived as follows [19-20]:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (1)$$

Fig. 3. Dynamic gesture with the command sequence: ‘start → switch → down → stop’

Fig. 4. Dynamic gesture with the command sequence: ‘start → volume → up → stop’

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (4)$$

$$\begin{aligned} \phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) & \left[\begin{aligned} & (\eta_{30} + \eta_{12})^2 - \\ & 3(\eta_{21} + \eta_{03})^2 \end{aligned} \right] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) & \left[\begin{aligned} & 3(\eta_{30} + \eta_{12})^2 \\ & -(\eta_{21} + \eta_{03})^2 \end{aligned} \right] \end{aligned} \quad (5)$$

$$\phi_6 = (\eta_{20} - \eta_{02}) \left[\begin{aligned} & (\eta_{30} + \eta_{12})^2 - \\ & (\eta_{21} + \eta_{03})^2 \end{aligned} \right] + \quad (6)$$

$$4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\begin{aligned} \phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) & \left[\begin{aligned} & (\eta_{30} + \eta_{12})^2 \\ & -3(\eta_{21} + \eta_{03})^2 \end{aligned} \right] \\ - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) & \left[\begin{aligned} & 3(\eta_{30} + \eta_{12})^2 \\ & -(\eta_{21} + \eta_{03})^2 \end{aligned} \right] \end{aligned} \quad (7)$$

Where η_{pq} is the normalised central moments defined by:

$$r = [(p + q) / 2] + 1$$

$$p + q = 2, 3, \dots$$

It is possible that using few features than seven values will still provide a very useful set of features. Fig. 5 shows different orientations of letter 'A' and their moment invariant features are shown in Table 1. We have used first four features in our system to classify for hand gestures using a Support Vector Machine (SVM) that will be discussed in next section.

III. SUPPORT VECTOR MACHINE (SVM) CLASSIFICATION

In our previous attempts, we successfully used Neural Networks for classification. We have now departed from that trend to SVM as it is widely used for statistical classification and regression analysis. The SVM classifier learns from data points in examples when they are classified belonging to their respective categories. SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite

Fig. 5. Letter 'A' in different orientations

TABLE I
MOMENT INVARIANTS FOR THE ABOVE 'A'

	A1	A2	A3	A4
ϕ_1	0.2165	0.2165	0.204	0.25153
ϕ_2	0.001936	0.001936	0.001936	0.002161
ϕ_3	3.6864e-005	3.6864e-005	3.6864e-005	0.004549
ϕ_4	1.6384e-005	1.6384e-005	1.6384e-005	0.002358
ϕ_5	-4.0265e-010	-4.0265e-010	-4.0265e-010	7.59E-06
ϕ_6	7.209e-007	7.209e-007	7.209e-007	7.11E-05
ϕ_7	0	0	0	1.43E-06

dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training datapoints of any class, since in general the larger the margin the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite dimensional space, it often happens that in that space the sets to be discriminated are not linearly separable. For this reason it was proposed that the original finite dimensional space be mapped into a much higher dimensional space presumably making the separation easier in that space. SVM schemes use a mapping into a larger space so that cross products may be computed easily in terms of the variables in the original space making the computational load reasonable. The cross products in the larger space are defined in terms of a kernel function which can be selected to suit the problem. The hyperplanes in the large space are defined as the set of points whose cross product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters of images of feature vectors which occur in the data base.

In our approach, Φ_1 - Φ_4 features are used as inputs into

The system now runs on any machine that runs Microsoft .net environment unlike our previous system which required Matlab running some of the latest tool boxes. This is a further boost to our quest to implement the whole system on a ‘Smart Phone’ currently available in the market so that gesture recognition can be very easily available for any equipment housing a camera. The system captures images every 250 milliseconds. The gestures that follow ‘start’ needs to be made consciously such that it will not cause blur or smearing. We are also conducting research into running entire processing on Field-programmable gate array (FPGA) bypassing a computer. This will very soon result in a Dynamic Hand Gesture System on a Chip (IC).

REFERENCES

- [1] P. Premaratne and Q. Nguyen, Consumer Electronics Control System based on Hand Gesture Moment Invariants, *IET Computer Vision*, Vol. 1-1, pp. 35-41, 2007.
- [2] S. Hutcheon, Last Hurrah for Lost Remote, *Sydney Morning Herald*, 18/07/2007, Web Link: <http://www.smh.com.au/articles/2007/07/18/1184559833067.html>
- [3] International Reporter, 16/07/2007, Web Link: <http://www.internationalreporter.com/News-2402/Now,-seven-simple-hand-gestures-to-switch-your-TV-on.html>
- [4] T. Baudel, and M. Baudouin-Lafon, Charade, Remote control of objects using free-hand gestures, *Comm. ACM*, 36, (7), pp. 28-35, 1993.

Fig. 6. Support Vector Machine input output relationship

the SVM as shown in Fig. 6. The weights are calculated using training data set similar to training a neural network [1].

IV. RESULTS AND DISCUSSION

The results of the proposed system were more than encouraging. We have managed to implement a dynamic gesture recognition system that recognized dynamic gestures that clearly identify ‘start’ and ‘stop’ function in real time.

- [5] S. S. Fels, and G. E. Hinton, Glove-Talk: A neural network interface between a data-glove and a speech synthesizer, *IEEE Trans. Neural Networks*, 4, pp. 2-8, 1993.
- [6] D.L. Quam, Gesture recognition with a dataglove, *Proc. 1990 IEEE National Aerospace and Electronics Conf.*, 2, pp. 755-760, 1990.
- [7] D. J. Sturman, and D. Zeltzer, A survey of glove-based input, *IEEE Computer Graphics and Applications*, 14, pp. 30-39, 1994.
- [8] C. Wang, and D. J. Cannon, A Virtual end-effector pointing system in point-and-direct robotics for inspection of surface flaws using a neural network-based skeleton transform, *Proc. IEEE Int'l Conf. Robotics and Automation*, 3, pp. 784-789, 1993.

- [9] R. Cipolla, Y. Okamoto, and Y. Kuno, Robust structure from motion using motion parallax', *Proc. IEEE Int'l Conf. Computer Vision*, pp. 374-382, 1993.
- [10] J. Davis, and M. Shah, Recognizing hand gestures, *Proc. European Conf. Computer Vision*, Stockholm, pp. 331-340, 1994.
- [11] Y. Kuno, M. Sakamoto, K. Sakata, and Y. Shirai, Vision-based human computer interface with user centred frame, *Proc. IROS'94*, 1994.
- [12] J. Lee, and T. L. Kunii, Model-based analysis of hand posture, *IEEE Computer Graphics and Applications*, pp. 77-86, 1995.
- [13] C. Maggioni, A novel gestural input device for virtual reality, *1993 IEEE Annual Virtual Reality Int'l Symp.*, pp. 118-124, 1993.
- [14] L. K. Lee, S. Ki, Y. Choi, M. H. Lee, Recognition of hand gesture to human-computer interaction, *IEEE 26th Annual Conf.*, 3, pp. 2117-2122, 2000.
- [15] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, P. Kiatisevi, Y. Shirai, and H. Ueno, Gesture based human-robot interaction using a frame based software platform, *IEEE International Conference on Man and Cybernetics*, 3, pp. 2883-2888, 2004.
- [16] M. Zobl, M. Geiger, B. Schuller, M. Lang, and G. Rigoll, A real-time system for hand gesture controlled operation of in-car devices, *Proc. Int. Con. Multimedia and Expo*, 3, pp. 541-544, 2003.
- [17] P. R.G. Harding, and T. Ellis, Recognizing hand gesture using Fourier descriptors, *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 3, pp. 286 – 289, 2004.
- [18] R. C. Gonzalez, R.E. Woods, S.L. Eddins, Digital image processing using MATLAB, (New Jersey, Pearson Prentice Hall, 2004).
- [19] Q. Zhongliang, and W. Wenjun, Automatic ship classification by superstructure moment invariants and two-stage classifier, *ICCS/ISITA '92 Communications on the Move*, pp. 544-547, 1992.
- [20] P. Premaratne, ISAR ship classification; An alternative approach, *CSSIP-DSTO Internal Publication*, Australia, March, 2003.