1-3-2007

Consumer electronics control system based on hand gesture moment invariants

Prashan Premaratne
University of Wollongong, prashan@uow.edu.au

Q. Nguyen
Australian National University

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

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Premaratne, Prashan and Nguyen, Q.: Consumer electronics control system based on hand gesture moment invariants 2007.
<https://ro.uow.edu.au/infopapers/605>

Consumer electronics control system based on hand gesture moment invariants

Abstract

Almost all consumer electronic equipment today uses remote controls for user interfaces. However, the variety of physical shapes and functional commands that each remote control features also raises numerous problems: the difficulties in locating the required remote control, the confusion with the button layout, the replacement issue and so on. The consumer electronics control system using hand gestures is a new innovative user interface that resolves the complications of using numerous remote controls for domestic appliances. Based on one unified set of hand gestures, this system interprets the user hand gestures into pre-defined commands to control one or many devices simultaneously. The system has been tested and verified under both incandescent and fluorescent lighting conditions. The experimental results are very encouraging as the system produces real-time responses and highly accurate recognition towards various gestures.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as Premaratne, P, and Nguyen, Q, Consumer electronics control system based on hand gesture moment invariants, IET Computer Vision, 1(1), 2007, 35-41. Copyright 2007 IET.

Consumer electronics control system based on hand gesture moment invariants

P. Premaratne and Q. Nguyen

Abstract: Almost all consumer electronic equipment today uses remote controls for user interfaces. However, the variety of physical shapes and functional commands that each remote control features also raises numerous problems: the difficulties in locating the required remote control, the confusion with the button layout, the replacement issue and so on. The consumer electronics control system using hand gestures is a new innovative user interface that resolves the complications of using numerous remote controls for domestic appliances. Based on one unified set of hand gestures, this system interprets the user hand gestures into pre-defined commands to control one or many devices simultaneously. The system has been tested and verified under both incandescent and fluorescent lighting conditions. The experimental results are very encouraging as the system produces real-time responses and highly accurate recognition towards various gestures.

1 Introduction

Human–computer interaction (HCI) has become an increasingly important part of our lives because of massive technological infusion into our lifestyles. Whether it is our living room, bedroom or office room, there could be a range of electronic equipment that needs 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 these items using voice recognition or glove-based devices [1–5] with mixed results. Glove-based devices are tethered to the main processor with cables which restricts the user's natural ability to communicate. Many of those 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/fingers [6–13].

Our research distinguishes from the previous attempts because of few marked differences:

- (i) a minimum number of gestures are used to offer higher accuracy with less confusion;
- (ii) only low processing power is required to process the gestures, which is useful for simple consumer control devices;
- (iii) very robust to lighting variations;
- (iv) real-time operation.

The desire to develop a limited set of hand gestures that are distinctive has improved the processing accuracy of

captured gestures with less computing power. This also requires a less sophisticated classification system using neural networks that does not need much processing power to work in real-time. The system has been thoroughly tested under both incandescent and fluorescent lighting to simulate home environments. It also incorporates text overlaid feedback to restrict the system responding to unintentional hand movements.

2 Hardware design overview

The system comprises a web camera, gesture processing unit, hardware interface for the control unit and a universal remote control. The webcam is used to capture the hand gestures which are then registered, normalised and feature-extracted for eventual classification to control the remote controller. The setup of the basic components is shown in Fig. 1. Matlab is used throughout the project for real-time data processing and classification and controlling through a parallel port. Once the user hand gesture matches with a pre-defined command, the command will be issued to the corresponding remote control via a parallel port. If an unknown gesture is issued, the system rejects it, notifying the user.

The interface circuitry is used to map parallel-port commands to the universal remote control. Four of the personal computer (PC) parallel-port data pins are used for multiplexing that achieves 16 unique controllable switches for controlling the remote controller. The block diagram of the hardware interface circuit and the photo of the fabricated circuit are shown in Fig. 2.

3 Gesture registration

The captured hand gestures from a real-time video stream need to be processed before they can be interpreted by a computer. It is extremely important that the captured image is registered as a hand gesture using skin segmentation after removing the background of the image. The skin segmentation techniques used in this research involves converting the image from RGB format to YCbCr format [14]. Then a

Fig. 1 Schematic representation of basic components

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. Thus it makes the segmentation less dependent on the lighting condition, which has always been a critical obstacle for image recognition. The threshold values were obtained using our own data set.

To find the characteristics of the pixels representing the skin region, a random image is selected out of many such images. The pixels’ properties are categorised to determine the threshold limits for the filter. Then, the converted picture in YCbCr format is viewed in the ‘imtool’ of MATLAB so that every pixel and its associated values such as x , y coordinates and intensity can be determined accurately. A number of sample points that represent skin patches and non-skin patches are obtained.

According to Fig. 3, both of the ‘skin’ pixels and the ‘non-skin’ pixels have their luminosity values spreading over the full range, that is, the Y component of skin patch is from 110 to 165, whereas that of non-skin patch is from 16 to 208. This property of the Y component implies that we are not filtering the luminance of the image, but the other two remaining components, Cb and Cr. For better visualisation, the Cb and Cr values of both skin and non-skin patches are plotted in a graph to find the region that they are likely to fall in. As can be seen from Fig. 3, the ‘skin’ pixels clearly distinguish themselves from the ‘non-skin’ ones. The thresholds of the filter can hence be determined easily. A similar approach is repeated on another image in fluorescent lighting conditions. The threshold values, as expected, are quite different as depicted in Table 1.

A number of images are used to evaluate effectiveness of the classification system. Both the original image and the skin-segmented image are used to observe and verify the accuracy of the segmentation algorithm. Images with low

Fig. 3 Luminosity values of skin and non-skin pixels

lighting condition are also tested. It is quite apparent from Fig. 4 that there are always a number of noisy spots in the filtered images, regardless of the lighting condition. This distortion, as expected, becomes more pronounced in low lighting conditions. As a result, the skin-segmented image is noisy and distorted and is likely to result in incorrect recognition at the subsequent stages. These distortions, however, can be removed during the gesture normalisation stage.

3.1 Gesture normalisation

Gesture normalisation is done by the well-known morphological filtering technique, erosion combined with dilation [15]. The output of this stage is a smooth region of the hand figure, which is stored in a logical bitmap image as shown in Fig. 5.

The experiments are carried out on an average computer of 1.6 GHz with 256 MB RAM. This is mainly to determine whether the system can operate from a set-top box with limited processing power. The observed execution time of 0.2 s is acceptable as it consumes only one-fifth of the available processing time (1 s). Shorter execution time can be obtained on a computer with better specification. Above all, when the system is implemented into a single integrated circuit (IC), hardware-based processing will be swifter.

3.2 Skin segmentation testing

This test was mainly aimed at evaluating the performance of the skin segmentation and normalisation modules of the control system. A number of hand gesture images were

Fig. 2 Block diagram of the hardware interface circuit and photo of the fabricated circuit

Table 1: Threshold values of incandescent and fluorescent lighting conditions

Incandescent	Fluorescent
$100 \leq Cb \leq 122$	$114 \leq Cb \leq 128$
$132 \leq Cr \leq 150$	$140 \leq Cr \leq 158$

taken and the skin segmentation and the subsequent normalisation are shown in Fig. 6. As seen from Fig. 6, the filter had successfully segmented the skin regions out of all the tested images. It was also noticeable that the shadow of the hand and the body did not have any effect on the filtering process. The remaining noise and unfilled pixels were removed by the normalisation filter which resulted in a smooth and clear region.

Non-uniform background images were also tested and some of the results are shown in Fig. 7. It was quite apparent from Fig. 7 that non-uniform background images produced many scattered patterns and noisy spots during the skin segmentation process. In particular, in the first two images taken under incandescent light, the hand was segmented along with some parts of the guitar and the edges of the wardrobe, forming a fairly distorted image. However, after being passed through the normalisation filter, the resultant images only consisted of the largest region found in filtered images, in this case effectively the hand region. This also implied that larger region of ‘skin-like’ objects might result in incorrect segmentation and should be carefully considered. The last two images taken under fluorescent light, on the other hand, showed significantly less noise than the first two images. This could be explained in terms of the difference in physical characteristics of the two light sources. In particular, incandescent light generated a yellowish glow which modified the look of objects that was captured. The resultant effect was that the object might have been recognised as a skin region as its colour had been modified. Fluorescent light, conversely, generated white light and thus retained the original colour of the object. This resulted in less ‘skin-like’ regions in the image and the hand could be extracted out more accurately. In conclusion, the performance of the skin segmentation and normalisation filters was firmly robust against the variance in backgrounds and lighting conditions. Nevertheless, the user should consider the negative effect created by

Fig. 4 Noisy spots in the filtered images

Fig. 5 Smooth region output of the morphological filtering technique

‘skin-like’ regions in the working environment under incandescent light.

4 Feature extraction

It is not too difficult to realise that effective real-time classification cannot be achieved using attempts such as template matching [16]. Template matching itself is very much prone to error when a user cannot reproduce an exact hand gesture to a gesture that is already stored in the library. It also fails because of variance to scaling as the distance to the camera may produce a scaled version of the gesture. The gesture variations because of rotation, scaling and translation can be circumvented using a set of features that are invariant to these operations. Moment invariants offer a set of features that encapsulate these properties.

4.1 Moment invariants

The moment invariants algorithm has been known as one of the most effective methods to extract descriptive features for object recognition applications. The algorithm has been widely applied in classification of aircraft, ships, ground targets and so on [17, 18]. 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

Fig. 6 Hand gesture images and the skin segmentation and the subsequent normalisation

Fig. 7 Results of the non-uniform background images

central moments of order $(p + q)$ of $f(i, j)$ are defined as

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

$$U_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j) \quad (2)$$

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 moment invariants are derived as follows [18]

$$\phi_1 = \eta_{20} + \eta_{02} \quad (3)$$

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

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (5)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (6)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (7)$$

$$\begin{aligned} \phi_6 = & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (8)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\ & \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (9)$$

where η_{pq} is the normalised central moments defined by

$$\eta_{pq} = \frac{U_{pq}}{U_{00}^r}$$

where $r = [(p + q)/2] + 1$ and $p + q = 2, 3, \dots$

Fig. 8 Rotated, scaled, translated and noisy versions of letter 'A'

4.2 Example of invariant properties

Fig. 8 shows images containing letter 'A', rotated and scaled, translated and noisy versions of letter 'A'. Their respective moment invariants calculated using formulas using (1)–(9) are shown in Table 2.

It is obvious from Table 2 that the algorithm produces the same result for the first three orientations of letter 'A' despite the different transformations applied upon them. There is only one value, that is, Φ_1 , displays a small discrepancy of 5.7% because of the difference in scale. The other values of the three figures are effectively the same for Φ_2 , Φ_3 , Φ_4 , Φ_5 , Φ_6 and Φ_7 . The last letter, however, reveals the drawback of the algorithm: it is susceptible to noise. Specifically, the added noisy spot in the letter has changed the entire moment invariants set. This drawback suggests that moment invariants can only be applied on noise-free images in order to achieve the best results. As the algorithm is firmly effective against transformations, a simple classifier can exploit these moment invariants values to differentiate as well as recognise the letter 'A' from other letters, such as the letter 'L' as follows (Fig. 9 and Table 3).

4.3 Application

The example has proved that moment invariants can be used for object recognition applications as they are rigidly invariant to scale, rotation and translation. The following account summarises the advantages of the moment invariants algorithm for gesture classification.

For each specific gesture, moment invariants always give a specific set of values. These values can be used to classify the gesture from a sample set. The set of chosen gestures have a set of unique moments.

- Moment invariants are invariant to translation, scaling and rotation. Therefore the user can issue commands disregarding orientation of his/her hand.
- The algorithm is susceptible to noise. Most of this noise, however, is filtered at the gesture normalisation stage.
- The algorithm is moderately easy to implement and requires only an insignificant computational effort from

Table 2: Moment invariants calculated using formulas (1)–(9) for letter '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	3.69×10^{-5}	3.69×10^{-5}	3.69×10^{-5}	0.004549
Φ_4	1.64×10^{-5}	1.64×10^{-5}	1.64×10^{-5}	0.002358
Φ_5	-4.03×10^{-10}	-4.03×10^{-10}	-4.03×10^{-10}	7.59×10^{-6}
Φ_6	7.21×10^{-7}	7.21×10^{-7}	7.21×10^{-7}	7.11×10^{-5}
Φ_7	0	0	0	1.43×10^{-6}

Fig. 9 Simple classifier exploiting moment invariants values to differentiate and recognise letter 'L'

the CPU. Feature extraction, as a result, can be progressed rapidly and efficiently.

- The first four moments, Φ_1 , Φ_2 , Φ_3 , and Φ_4 are adequate to represent a gesture uniquely and hence result in a simple feature vector with only four values.

5 Gesture classification

Having accomplished all the above stages, we have successfully extracted a data set from an image of a user hand gesture. However, this data set remains meaningless unless the program can interpret it into a preset command to control the electronic device. The classification process, thereby, can be done either by a nearest-neighbour classifier or via a neural network.

On one hand, both the methods require a training set containing a number of sample images. After the feature extraction stage, each group of the sample images that represent the same gesture produces a certain range of Φ_1 , Φ_2 , Φ_3 and Φ_4 . These ranges are then used as preset values to classify a random input image. The procedure implicitly states that the more samples we have, the better the classification becomes.

On the other hand, the nearest-neighbour classifier is more computationally intensive than the neural network. In particular, the former approach involves calculating the distance from a new point to all of the points in the sample set. The value of the new point is then rounded to that of the sample point which produces the minimum distance. Therefore the more values in the sample set, the longer it takes to compute and determine the output, especially when the system complexity is elevated. A statistical approach can be applied to determine a small number of prototypes out of a sample set. However, it is extremely time-consuming to analyse the method and therefore not practically feasible to achieve real-time operation.

A neural network classifier, however, proves itself to be more effective and more efficient. Neural networks have been applied to perform complex functions in numerous applications, including pattern recognition, classification, identification and so on. Once implemented, they can compute the output significantly quicker than the nearest-

neighbour classifier. Neural networks also encompass the ability to learn and predict over the time. This property enables the system to be viewed more as a human-like entity that can actually 'understand' the user, which is also one of the major objectives of our research.

5.1 Proposed neural network design

The system is designed to capture one image frame (static image) every second and is then segmented for skin region detection and other pre-processing before the invariant moments are calculated. These invariant moments will be the input to the neural network for classification and the subsequent action using the remote control and the feedback system. If any of the static images is captured when the hand is moving, the resultant image would be blurred. This will result in an unrecognised hand gesture and the user will be informed about it through the system feedback display.

The designed neural network is a 'backpropagation' network, in which input vectors (invariant moments of the sample set of user hand gestures) and the corresponding target vectors (the commands set) are used to train the network until it can approximate a function between the input and the output [19–21]. In this particular design, there are only three layers because of the limited number of hand gestures to be classified. More complex networks could be possibly designed and implemented, but it is neither practical nor necessary for our research. For better visualisation, the network can be illustrated in Fig. 10 where W represents the weighting function in which each input is weighted with an appropriate w , and b represents the bias coefficient and it is set to 1 in this design.

5.1.1 Evaluation of the best set of gestures: One of the main goals of the project was to identify a set of gestures that would optimise the classification, being able to operate most of the functions of a remote control. It was decided that an initial set of 20 hand gestures be selected and be pruned overtime after evaluating their classification scores. The initial neural network classifier was trained with 995 samples representing 20 gestures (each gesture approximately had 50 different images) and each gesture was evaluated five times using another five instances of each gesture. Some of these gestures and their classification scores are tabulated in Fig. 11. Gestures with poor classification scores were eliminated one at a time and the system was trained on the remaining gesture set and was re-evaluated. This process eventually resulted in best seven gestures and six of them are shown in Fig. 12.

5.1.2 System training using the optimum set: After the elimination process described in the above section, the system was retrained on the new optimum (this was an

Table 3: Moment invariants calculated using formulas (1)–(9) for letter 'L'

	L1	L2	L3
Φ_1	0.34028	0.31944	0.31944
Φ_2	0.043403	0.043403	0.043403
Φ_3	0.023148	0.023148	0.023148
Φ_4	0.002572	0.002572	0.002572
Φ_5	-5.56×10^{-6}	-5.56×10^{-6}	-5.56×10^{-6}
Φ_6	-0.00015	-0.00015	-0.00015
Φ_7	1.91×10^{-5}	1.91×10^{-5}	1.91×10^{-5}

Fig. 10 Network illustrating better visualisation

Hand gesture
Classification accuracy	44%	50%	74%	82%	97%	90%

Hand gesture
Classification accuracy	95%	96%	70%	95%	47%	89%

Fig. 11 Gestures and their classification accuracy

observation) set using 300 sample values of seven gestures, and 300 corresponding outputs are used to train the network. The iteration limit is set to 500 times and the mean square error (MSE) is set to 0.05. These limits ensure that the network has a sufficient set of training data to develop an accurate transfer function between the input and the output. Furthermore, the network is trained twice to improve the accuracy and the precision of the transfer function. The output of the above code is a 10×1 vector named 'label', in which there is only one number 1 and the rest is 0. The index of the only number 1 is also the command that the input gesture is interpreted. For instance, $\text{label} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ indicates command number 3; $\text{label} = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ indicates command number 5. These commands are then transferred to interface circuit to control the device. A complete table of the command codes is listed in Table 4.

5.2 Test results

The experiments were carried out on a computer featuring a 1.6 GHz processor with 256 MB of RAM running MATLAB 7.01 SP1. A software program RCS was written to display the feedback to the user as well as to display command being implemented when the hardware is controlled. A Panasonic CRT television and a VCR were used for the experiments. Currently the system needs few seconds to analyse the user's hand in order to determine the threshold value for skin segmentation and store it. The first gesture needed to initialise the hardware is the 'Start' followed by 'Power-On' gesture. This can be followed by VCR or TV selection. Even though only two consumer

Fig. 12 Six of the best seven gestures

Table 4: Complete table of commands codes

Button	Remote control code	Command code
TV1	1	8
Video1	1	7
Volume Up	1	2
Volume Down	1	3
Channel Up	1	9
Channel Down	1	6
Power	1	1
TV/AV	1	5
Play	1	4
Stop	1	10

electronics devices are used, any number of devices can be controlled. Any command can be issued randomly; however, if they are not issued in a logical manner, a proper course of action cannot be taken. For instance, if 'Up' or 'Down' command is issued prior to 'Volume' or 'Channel', even though the command is recognised, no action will be taken. The system was observed to be 100% accurate under normal lighting conditions for both fluorescent and incandescent lights.

The tests have firmly consolidated the hardware design and the software interface of the developed prototype. The hardware module produces a very fast response to the outputs of the parallel port as well as delivering correct commands to the remote control. Different hand gestures along with their gesture extractions are shown in Fig. 12.

6 Conclusions and future work

The system is developed to reject unintentional and erratic hand gestures (such as children's irrational movements) and to supply visual feedback on the gestures registered. This work managed to invent a set of gestures that are distinct from each other yet easy to recognise by the system. This set has unique four invariant moments which result in highly accurate and real-time classification. The accuracy of the control system was 100% and was mainly because of the limited number of hand gestures. This set of hand gestures is adequate for any consumer electronic control system. The software interface produces unique key mapping ability such that the 'Volume' gesture in TV mode can be mapped to the 'Speed' function of a ceiling fan. In future, we expect to utilise an IR camera to address poor lighting conditions. This system is currently ready to be implemented on dedicated hardware such as a digital TV set-top box.

7 Acknowledgment

The authors thank Vimarshana Bandaranayake of the University of Wollongong for her support in conceptual development of the gesture control system.

8 References

- Baudel, T., and Baudouin-Lafon, M.: 'Charade: remote control of objects using free-hand gestures', *Commun. ACM*, 1993, **36**, (7), pp. 28–35
- Fels, S.S., and Hinton, G.E.: 'Glove-Talk: a neural network interface between a data-glove and a speech synthesizer', *IEEE Trans. Neural Netw.*, 1993, **4**, pp. 2–8
- Quam, D.L.: 'Gesture recognition with a dataglove'. Proc. 1990 IEEE National Aerospace and Electronics Conf., 1990, vol. 2, pp. 755–760

- 4 Sturman, D.J., and Zeltzer, D.: 'A survey of glove-based input', *IEEE Comput. Graphics Appl.*, 1994, **14**, pp. 30–39
- 5 Wang, C., and Cannon, D.J.: '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. Conf. Robot. Automation, 1993, vol. 3, pp. 784–789
- 6 Cipolla, R., Okamoto, Y., and Kuno, Y.: 'Robust structure from motion using motion parallax'. Proc. IEEE Int. Conf. Comput. Vision, 1993, pp. 374–382
- 7 Davis, J., and Shah, M.: 'Recognizing hand gestures'. Proc. European Conf. Computer Vision, Stockholm, 1994, pp. 331–340
- 8 Kuno, Y., Sakamoto, M., Sakata, K., and Shirai, Y.: 'Vision-based human computer interface with user centred frame'. Proc. IROS'94, 1994
- 9 Lee, J., and Kunii, T.L.: 'Model-based analysis of hand posture', *IEEE Comput. Graphics Appl.*, 1995, pp. 77–86
- 10 Maggioni, C.: 'A novel gestural input device for virtual reality'. 1993 IEEE Annual Virtual Reality Int. Symp., 1993, pp. 118–124
- 11 Lee, L.K., Ki, S., Choi, Y., and Lee, M.H.: 'Recognition of hand gesture to human–computer interaction'. IEEE 26th Annual Conf., 2000, vol. 3, pp. 2117–2122
- 12 Hasanuzzaman, M., Zhang, T., Ampornaramveth, V., Kiatisevi, P., Shirai, Y., and Ueno, H.: 'Gesture based human–robot interaction using a frame based software platform'. IEEE Int. Conf. Man Cybernet., 2004, vol. 3, pp. 2883–2888
- 13 Zobl, M., Geiger, M., Schuller, B., Lang, M., and Rigoll, G.: 'A real-time system for hand gesture controlled operation of in-car devices'. Proc. Int. Con. Multimedia and Expo, vol. 3, pp. 541–544
- 14 Marius, D., Pennathur, S., and Rose, K.: 'Face detection using colour thresholding, and Eigen image template matching', EE368: Digital Image Processing Project, Stanford University, Stanford, CA, May 2003. http://www.stanford.edu/class/ee368/Project_03/Project/reports/ee368group15.pdf, accessed 20, August 2003
- 15 Gonzalez, R.C., Woods, R.E., and Eddins, S.L.: 'Digital image processing using MATLAB' (Pearson Prentice-Hall, 2004)
- 16 Shan, C., Wei, Y., Qiu, X., and Tan, T.: 'Gesture recognition using temporal template based trajectories'. Proc. 17th Int. Conf. Pattern Recognition, 2004, vol. 3, pp. 954–957
- 17 Zhongliang, Q., and Wenjun, W.: 'Automatic ship classification by superstructure moment invariants and two-stage classifier'. ICCS/ISITA'92 Communications on the Move, 1992, pp. 544–547
- 18 Premaratne, P.: 'ISAR ship classification: an alternative approach' (CSSIP-DSTO Internal Publication, Australia, 2003)
- 19 Fausett, L.: 'Fundamentals of neural networks' (Prentice-Hall, 1994)
- 20 Gurney, K.: 'An introduction to neural networks' (UCL Press, 1997)
- 21 Haykin, S.: 'Neural networks' (Prentice-Hall, 1999, 2nd edn.)