J Supercomput (2013) 65:336–352
DOI 10.1007/s11227-010-0541-9

Vision-based arm gesture recognition for a long-range human–robot interaction

DoHyung Kim · Jaeyeon Lee · Ho-Sub Yoon · Jaehong Kim · Joochan Sohn

Published online: 4 January 2011

Abstract This paper proposes a vision-based human arm gesture recognition method for human–robot interaction, particularly at a long distance where speech information is not available. We define four meaningful arm gestures for a long-range interaction. The proposed method is capable of recognizing the defined gestures only with 320×240 pixel-sized low-resolution input images captured from a single camera at a long distance, approximately five meters from the camera. In addition, the system differentiates the target gestures from the users' normal actions that occur in daily life without any constraints. For human detection at a long distance, the proposed approach combines results from mean-shift color tracking, short- and long-range face detection, and omega shape detection. The system then detects arm blocks using a background subtraction method with a background updating module and recognizes the target gestures based on information about the region, periodical motion, and shape of the arm blocks. From experiments using a large realistic database, a recognition rate of 97.235% is achieved, which is a sufficiently practical level for various pervasive and ubiquitous applications based on human gestures.

Keywords Gesture recognition · Face detection · Omega shape detection · Long-range human–robot interaction

1 Introduction

In pervasive computing environments, vision-based gesture recognition technology is widely used with speech recognition for communication between a human and a multimedia system [1, 2]. Gesture recognition can be more useful, particularly at a long distance, compared to a short distance where speech information is available.

D. Kim (✉) · J. Lee · H.-S. Yoon · J. Kim · J. Sohn
Electronics and Telecommunications Research Institute, Daejeon, Korea
e-mail: dhkim008@etri.re.kr

Gesture recognition is not a new issue. Numerous studies on gesture recognition have been conducted [3, 4]. However, there are a number of limitations that arise when existing gesture technologies are adopted for long-range human–robot interaction.

- The existing gesture recognition systems do not provide meaningful gestures and a recognition method for human–robot interaction at a long distance. In robot environments, most technologies involve commanding robots at a short distance by means of recognized hand gestures [5]. Gesture recognition technologies at a long distance are mostly focused on human behavior awareness based on an analysis of the entire body of the human [6]. There have been few trials thus far regarding human gesture recognition for communication between a human and a robot at a long distance.
- Some existing studies that demonstrated remote recognition of the upper body or the entire human body and related gestures are inappropriate for low-cost robots with a single camera, as most require the use of high-priced equipment such as high-resolution cameras, stereo cameras, and 3D scanners [7].
- Other existing methods that recognize long-range gestures only with a single camera assume an environment with a fixed camera [8]. Hence, these systems are not feasible for mobile robots.
- For reliable operations, many gesture recognizers place several constraints on users [9]. For example, one recognizer asks users to wear red gloves or blue clothes. However, it is unrealistic to expect users to cooperate for recognition purposes in robot environments in which users are freely acting.

Therefore, this necessitates the development of a novel gesture recognizer that satisfies the requirements for long-range human–robot interaction. This paper proposes a vision-based human arm gesture recognition method for human–robot interaction at a long distance.

As shown in Fig. 1, we define four meaningful gestures for human–robot interaction at a long distance and set the target gestures of the proposed system.

- "*Waving*" is a motion gesture involving the waving of a person's right arm for approximately three seconds. A human can use the waving gesture to attract a robot's attention. When a user wants to initiate interaction with a robot located at a long distance, the user can use this gesture as a starting signal. Additionally, when a human wants to decline a service from a robot, "no" can also be expressed with this gesture.
- "*Calling*" is a gesture that involves the waving of a person's right arm up and down for approximately three seconds. The calling gesture may be used to summon a robot.

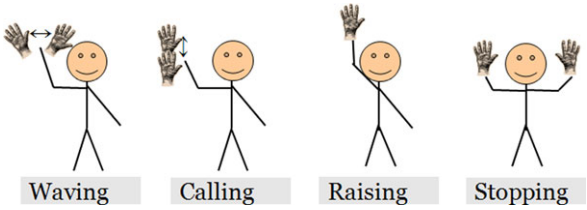

Fig. 1 Four target gestures for long-range human–robot interaction

- "*Raising*" is a motionless gesture that involves the holding of the right arm straight upwards for approximately three seconds. This gesture is defined for a human to show his/her ID indirectly or to signal "yes" when offered a proactive service by robot.
- "*Stopping*" is a motionless gesture that involves the user raising both arms about the head for approximately three seconds. A human can signal "stop" to robots by executing this gesture.

Generally, users will execute the defined gestures only when they want to express an intention to a robot. Consequently, the gesture recognition system essentially assumes that users are always looking directly at the robot when they execute these gestures. This assumption is not a constraint on the actions of the user. Facing each other is an initial step in the execution of gestures.

A gesture recognition system for robots requires the ability to distinguish target gestures from users' common actions, such as scratching their head, clasping their hands above their head, stretching their body with raised hands and other such gestures. This is an essential ability of a gesture recognizer in an actual home environment. The proposed system provides this functionality.

In human–robot interaction field, short distance means about within two meters where human and robot can interact with each other actively using speech, facial expression, hands gesture and so on. In this paper, the long distance means more than three meters where these communication methods are not available.

2 Configuration of the proposed system

To recognize the defined target gestures for long-range interactions, the proposed system follows the flow chart shown in Fig. 2.

When 320×240 pixel-sized low-resolution input images are fed into the system, a face detection module is initially triggered. The face detector is designed to detect a minimum of 12×12 pixel-sized faces located at a maximum range of five meters by effectively combining the results from a mean-shift color tracker, short- and long-range face detectors, and an omega shape detector.

The proposed system then determines a gesture search region that consists of five subregions based on the detected face region. As the four target gestures only appear in the established search region, the method examines if the target gestures appear only in this region.

When the system detects the user's face and determines the gesture search region successfully, a background image capturing process is triggered to detect arm blocks using a background subtraction method. Given that robots normally move around freely, we implemented a partial background updating module only for the gesture search region.

After segmenting arm blocks, the system finally recognizes the target gestures based on the information about the region, periodical motion, and shape of the arm blocks. We initially investigate the subregions that contain the detected arm blocks to remove noise blocks or users' normal actions that are regarded as noise gestures. Subsequently, to recognize motion gestures such as waving and calling, the system

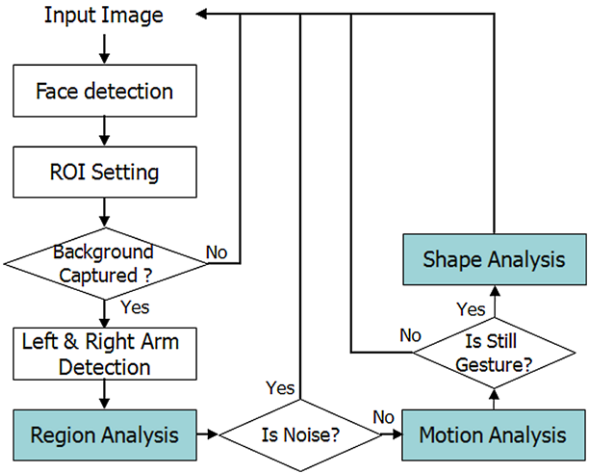

Fig. 2 Flow chart of the proposed system

analyzes the moving direction of the right arm block. If the right arm block moves periodically, the system assumes that the moving gestures occur. The gesture recognizer also examines the shape of the arm blocks, including the length, size, and position, in order to recognize the motionless gestures of raising and stopping.

3 Tiny face detection

Face detection is one of the most fundamental features of intelligent robots. Various face detection techniques have been reported over the years. Several recent methods have performed well enough to be included in the systems of electronic products such as digital cameras [10].

However, the capability of the face detector required in robot environments differs from that of traditional detectors. In robot environments, the robot must detect small human faces from a long distance in addition to larger faces, as interactions can occur regardless of the distance between a human and the robot. It is evident that faces should also be detected while the robot is moving. In addition, taking into account many low-cost robots, processing with low-resolution input images is strongly recommended.

Most existing methods can detect only large faces within a short range of less than three meters from the robot. Several studies involving the detection of tiny faces at a long distance make use of high-resolution input images, the zoom-functions of high-priced cameras, or fixed cameras that are not suitable for robots [11, 12]. One report described the detection of 6×6 pixel-sized minuscule faces; however, it showed a high false positive rate at the same time [13].

Therefore, we propose a practical detection method for tiny faces for mobile robots. The method facilitates the detection of faces as small as 12×12 pixels that are located at a maximum range of five meters, even with low-resolution images 320×240 pixels in size.

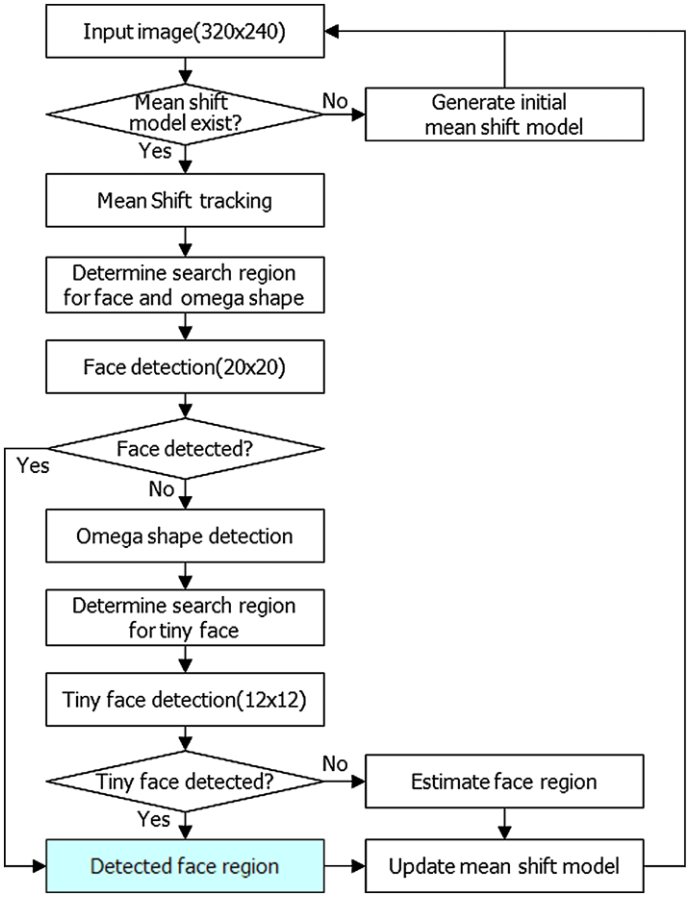

Fig. 3 Flow chart of the proposed method

We solve the tiny face detection problem by organizing a system that consists of multiple detectors. Included are a mean-shift color tracker, short- and long-range face detectors, and an omega shape detector, as shown in Fig. 3.

The first step of the method involves skin color tracking based on a mean-shift method [14]. If the initial face skin color model for the mean-shift tracker is not yet built, the system performs face detection using the input image. When a face region is located successfully, the color within the face region is used to generate the initial mean-shift model. Many researchers have adopted a skin color tracker in order to track faces. However, because the tracker uses color information, it is vulnerable to changes in lighting conditions. Therefore, the proposed system makes use of a mean-shift color tracker in order not to track faces directly, but to only determine search regions for the face detector.

After the face search region is determined by the mean-shift tracker, the face detector, which is designed to detect faces as small as 20×20 pixels, runs only within that region. The adopted face detection algorithm is implemented based on the AdaBoost

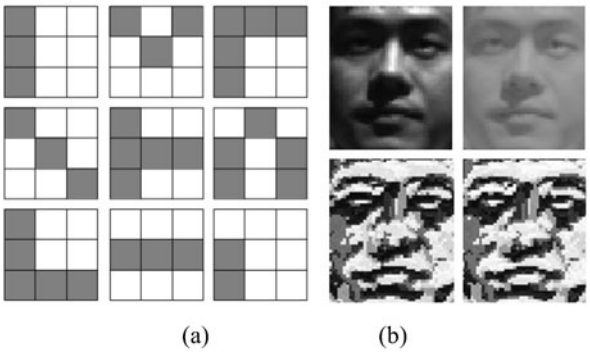

Fig. 4 Modified census transform: (**a**) example patterns of MCT, (**b**) example of MCT illumination invariance

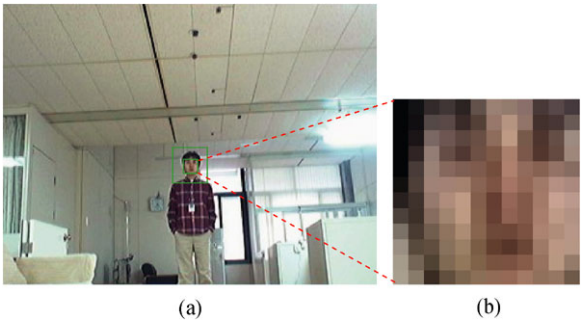

Fig. 5 An example of detected face at a long distance: (**a**) 320 × 240 pixel input image captured from a robot camera located at a distance of five meters; (**b**) 12 × 12 pixel detected face image

procedure [15]. In addition, a modified census transform (MCT) feature is used [16]. As shown in Fig. 4, MCT converts the pixel values into one of 511 patterns in a 3×3 neighborhood. These patterns represent the local spatial information of edges, junctions, and line segments. The transformed structure is robust in terms of illumination variation.

When human subjects are close to a robot, roughly within 3 meters, the designed face detector performs well. However, at greater distances between the subject and the robot and when the facial regions are smaller than 20×20 pixels, the face detector cannot locate the facial regions.

To solve this problem, a face detector for tiny faces was specially designed based on the AdaBoost method, which has the capability to detect faces as small as 12×12 pixels. As shown in Fig. 5, the 12×12 pixel size is too small a size to include all of the main facial components, such as the eyes, nose, and mouth; consequently, it is extremely difficult to detect such tiny faces reliably, even with moving cameras. That is, the detection system cannot avoid the frequent occurrence of false positives, even if the detector is designed well enough to detect such tiny face regions. In contrast, if the detector is designed to reduce the number of false positives, it has a difficult time detecting such small faces. Therefore, it is nearly impossible to solve the tiny face detection problem using only a single face detector, even when the detector is well trained.

We solve this problem by adopting a face detector for tiny faces that is trained well enough to detect faces at a long range. Its operation is limited to a search region that

is determined automatically. If the detection of faces over 20×20 pixels in size fails, omega shape detection is then performed only within the search region determined by the mean-shift tracker. An omega shape indicates the upper part of the body, which includes the head and shoulders. As the omega shape is a much simpler pattern compared to a face, the performance of the omega detector is somewhat worse than that of the face detector. However, an omega shape has an important advantage in that it is detectable at a long distance, as the physical size of the region is relatively large compared to the facial region. The proposed omega shape detector is trained using the same method used with the face detector. If an omega shape is found, the detected omega shape region is assigned to the search region of the tiny face detector.

Thus, by focusing on limiting the face search region by as much as possible, the proposed method can accurately detect tiny faces at a long distance and even with a low-resolution image. This sharply decreases the number of false positives.

4 Long-range gesture recognition

4.1 Arms detection and region analysis

After detecting a face, the proposed method determines a gesture search region that consists of five subregions based on the detected face region, as shown in Fig. 6. Because the four target gestures only appear in the established search region, the method examines if the target gestures appear only in the region.

When the system detects a user's face and determines the gesture search region successfully, a background image capturing process is triggered to detect arm blocks, as shown in Fig. 7, in conjunction with a background subtraction method. Given that robots normally move around freely, we implemented a partial background updating module only for the gesture search region. That is, the system replaces a background image with a current image if arm blocks do not exist in the search region. Therefore, this technology is applicable to a robot environment if the robot remains motionless only when a user makes a gesture, although the robot or the user may move freely. This condition is fully satisfied in most robot service environments.

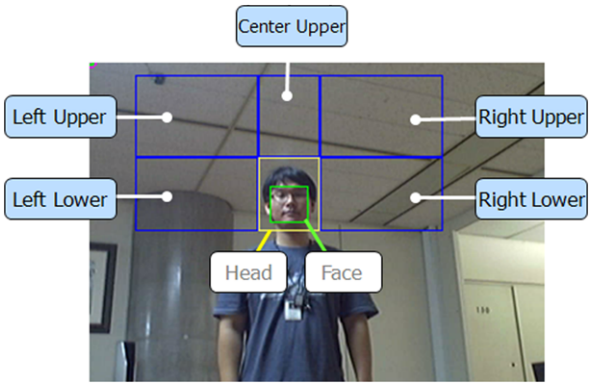

Fig. 6 Arm gesture search region

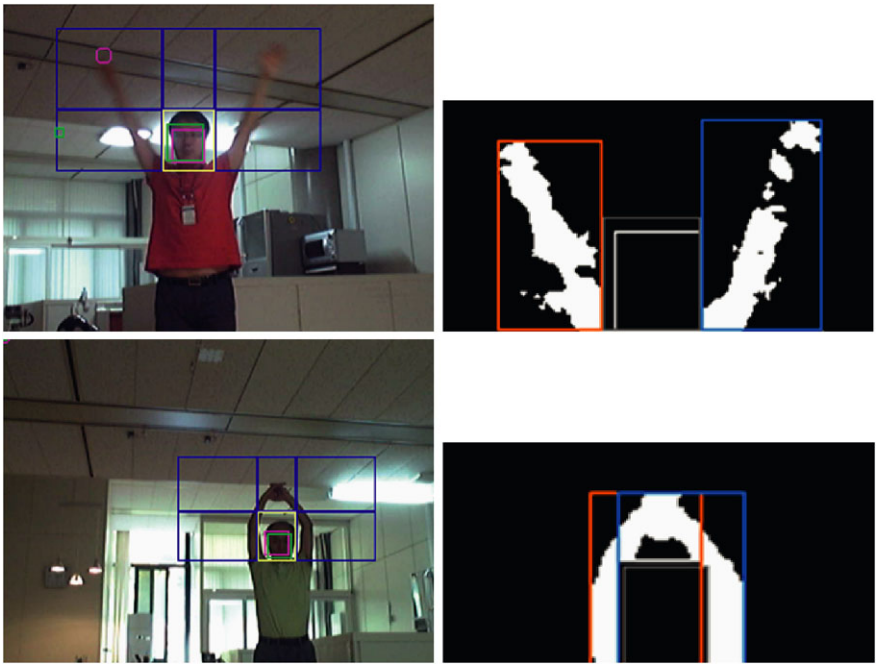

Fig. 7 Example of detected arm blocks

Table 1 Look-up table for the region analysis

	LL	LU	CU	RU	RL
Waving	o	–	–	x	x
Calling	o	–	–	x	x
Raising	o	o	–	x	x
Stopping	o	–	x	–	o

(o) Arm blocks must appear;
(x) Arm blocks must not appear;
(–) Does not matter

After segmenting the arm blocks, the system finally recognizes the target gestures based on the information pertaining to the region, the periodical motion, and the shape of the arm blocks.

We initially investigated the subregions with detected arm blocks. Table 1 shows a look-up table for the region analysis. Using the look-up table, we remove noise blocks or users' normal actions that are regarded as noise gestures. For example, when the human executes the "waving" gesture, an arm blob must appear in the LeftLower (LL) region without fail and must not appear in the RightUpper (RU) or the RightLower (RL) region. This blob can appear, or not, either in the LeftUpper (LU) region or in the CenterUpper (CU) region according to the style of action by the user. If a gesture cannot satisfy all of the conditions of the four target gestures described in the look-up table, it is determined as a noise gesture.

Therefore, the region analysis module cannot sufficiently recognize one of four gestures; it is therefore only responsible for filtering noise gestures, which are users'

common actions such as those that occur during the course of a normal daily. By implementing the region analysis module, it is possible to prevent misclassifications with minimal effort and to improve the efficiency of the system in terms of its computation cost.

4.2 Motion analysis

For the recognition of motion gestures such as waving and calling, the system analyzes the moving direction of the right arm block. If the right arm block moves periodically, the system assumes that the moving gesture occurred.

First, the system detects the coordinates of a fingertip from the region of the right arm as separated by the background subtraction process. The value of the y-coordinate of the fingertip is assigned as the value of the y-coordinate of the detected region of the right arm. The value of the x-coordinate of the fingertip is assigned as the value of the centroid of the x-coordinate in the upper one-fifth of the detected right arm region. Once the position of the fingertip is detected, determining the direction of the fingertip movement from the input image streams is straightforward.

After detecting the direction of the fingertip movement, the system analyzes the direction using a code table for movement directions as shown in Fig. 8 and an accumulated histogram of codes. The direction in which the fingertip moved is transformed to the value of a four-digit code. For example, the value of the code assigned is 1 if the fingertip moved to the right, because the movement direction is above a 315-degree angle and below a 45-degree angle. On the contrary, if the fingertip moved to the left, 3 will be assigned as the value of the code.

These code values of movement directions from the consecutive input images are continuously calculated and managed by an accumulated histogram for movement direction. The recognizer can detect whether the waving and calling gestures have occurred, as they are repetitive arm gestures that occur in a specific direction. For the waving gesture, the code values of 1 and 3 often occur in the accumulated histogram for movement direction in a normal case, as the waving gesture involves waving the

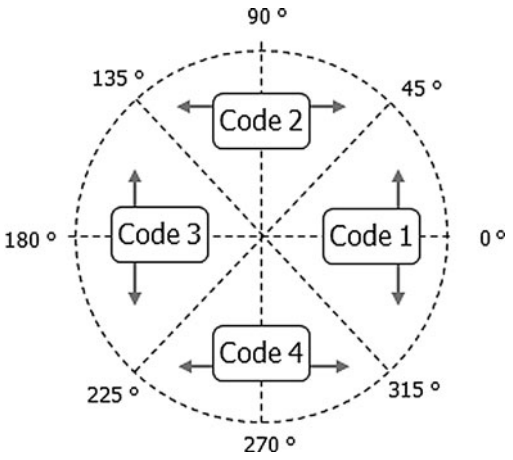

Fig. 8 Code table for movement direction

right arm repeatedly. Therefore, the recognizer determines that the waving gesture has occurred if code 1 or 3 is greater than the determined threshold of T1 and if code 3 or 1 corresponding to the opposite direction is above threshold T2. In other words, the recognizer can determine whether the waving or calling gesture has occurred by checking whether the code value for the opposite direction of the specific code is above threshold T2 when a specific code value is above threshold T1.

Additionally, the gesture recognizer creates the accumulated histogram by accumulating the weight W, not based on the code frequency but based on the movement velocity, as shown in (1).

$$W = 10\left(\frac{2\sqrt{(x_f - x_{f+1})^2 + (y_f - y_{f+1})^2}}{\sqrt{w_{LL}^2 + h_{LL}^2}} \right) \tag{1}$$

In (1), (x_f, y_f) and (x_{f+1}, y_{f+1}) represent the coordinates of the detected fingertip in the f-th image and $f+1$-th image, and w_{LL} and h_{LL} represent the width and height of the LeftLower (LL) region, respectively.

The proposed recognizer composes the accumulated histogram not by using the code frequency but by using the movement velocity so as to recognize users' gestures with an even response time regardless of the capturing speed of input images of various types of hardware equipment.

For example, the code values for movement direction decrease in frequency if the capturing speed of the input images is slow because the number of input images decreases during a certain period. In this case, a long time is required for an accumulated score to exceed threshold T1 if the gesture recognizer makes the accumulated histogram using the code frequency. Therefore, the recognizer requires more time to recognize users' gestures, and this causes users some inconvenience in that they have to repeat the same gesture for a long time. Alternatively, if the capturing speed of the hardware equipment is very fast, the response time of the recognizer can be very fast as code values are received in large quantities when a user waves his or her hand from side to side only once or twice. However, the possibility that the recognizer will misclassify common actions as waving gestures increases because the recognizer becomes excessively sensitive to the action velocity of users' gestures.

The proposed accumulated histogram using movement velocity as a weight value solves this problem. According to (1), if the displacement of the fingertip movement is higher within consecutive images, the weight W increases more. That is, if the capturing speed of the input images is relatively slow, the number of input images decreases, causing the frequency of the code to decrease. However, the weight W grows because the displacement of the fingertip movement increases within the consecutive images. Inversely, if the displacement is relatively low, the weight W will decrease more. Therefore, the proposed recognizer can recognize motion gestures in the event of even response times regardless of the capturing speed of the input images and irrespective of the user's gesture speed.

On the other hand, every weight W added to the proposed accumulated histogram has its own timestamp when it is added to the histogram. This is maintained in the histogram for a certain period of time (in this paper, for five seconds) and is then deleted.

4.3 Shape analysis

The gesture recognizer examines a shape of the arm blocks, including the length, size, and position, to recognize the motionless gestures of raising and stopping.

The raising gesture decision module can determine whether an action is a raising gesture or a noise gesture by simply analyzing the length and angle of the region of the right arm. The decision module determines that it is the raising gesture only if the length of the region of the right arm is 1.3 times longer than the height of the region of the user's head and if the direction of the stretched arms is between a 60-degree angle and a 135-degree angle. In other cases, the decision module determines that the gesture is a noise gesture.

The stopping gesture decision module can determine whether an action is a stopping gesture or a noise gesture by analyzing the length, angle, and position of the two regions of the left and right arms. First, the decision module determines that a gesture is a noise gesture if the two regions of both arms reach the region of the user's head, which pertains to a common daily action, or if the length of a region of a left or right arm is 1.6 times longer than the height of the region of the user's head.

Due to the definitions of stopping gesture actions and the characteristics of the structure of the human body making the gesture, there is a strong possibility of a stopping gesture if the angle of both hands is not more than 90 degrees. If the angle is more than 90 degrees, there is a strong possibility that the action is a common action, in other words, that it is not a stopping gesture, such as stretching. In addition, there is a strong possibility of a stopping gesture if the regions of both arms are extended outwards by as much as the width of the region of the user's head (Fig. 9).

Therefore, the system first calculates the possibility of a stopping gesture according to the angle and position of the region of the arms, as expressed below in (2), (3), and (4). If the final possibility (P) based on these possibilities is higher than a predetermined threshold, the system determines that the gesture is a stopping gesture. Otherwise, it determines that it is a noise gesture.

$$P_d = 1 - \frac{\mathrm{Max}(d, d_l) - d_l}{d_h - d_l} \tag{2}$$

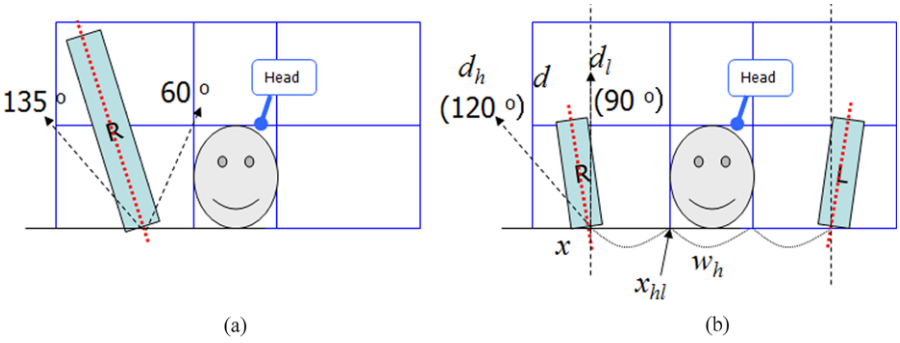

Fig. 9 Shape analysis for recognizing raising and stopping gestures: (**a**) conditions for the raising gesture; (**b**) conditions for the stopping gesture

$$P_p = \frac{\text{Min}(|x - x_{hl} + w_h|, w_h)}{w_h} \tag{3}$$

$$P = \alpha P_d + (1 - \alpha) P_p \tag{4}$$

Here P_d denotes the possibility of a stopping gesture based on the angle information of the region of the right arm, d represents the angle of the region of the right arm, and d_l and d_h represent the allowable degrees of a gesture (these values are 90 degrees and 120 degrees respectively). P_p is the possibility of a stopping gesture based on the position information of the region of the right arm, x represents the x-coordinate of the region of the right arm, x_{hl} represents the x-coordinate of the left border of the region of the user's head, and w_h represents the width of the region of the users' head. P is the final possibility of a stopping gesture and α is a weight between P_d and P_p. P_d and P_p are calculated only for the right arm because both arms of the stopping gesture are symmetrical to each other.

If a raising gesture or stopping gesture occurs more frequently than a predetermined threshold, the system lastly determines that the gesture has occurred.

5 Experimental results

In this section, we provide experimental results of the proposed system. The performance of the long-range gesture recognizer greatly depends on that of the face detector for tiny faces. Therefore, we evaluated this face detector using a large and realistic database.

5.1 Evaluation of the face detector

To train the detectors, we collected 25,060 face samples and 60,000 negative samples for the face detector, and 9666 omega shape samples and 50,000 negative samples for the omega shape detector. Figure 10 shows sample images for training of the face and omega shape detector. As a measure for the evaluation, the following recall and precision rates were used:

$$\text{recall rate} = \frac{N_c}{N_t} \tag{5}$$

$$\text{precision rate} = \frac{N_c}{N_c + N_f} \tag{6}$$

Here, N_c and N_f are the numbers of correctly detected faces and falsely detected faces, respectively, and N_t is the total number of faces in all frames.

To evaluate the performance of the system, we generated two different types of realistic databases. The first database was generated to evaluate the performance of the long-range face detector compared to the short-range detector. The database contained a total of 22,500 images captured from a range of one to five meters in normal home environments. The human subjects were requested to look at the robot and to move their faces slightly to the left and right. The other database was generated to

Fig. 10 Sample images for training of the face and omega shape detector: (**a**) face images, (**b**) omega shape images, (**c**) negative images

determine how the system would operate in an actual robot environment. A moving robot recorded eight indoor and outdoor video clips. The subjects moved around freely, ignoring the robot while maintaining a distance of three to five meters. In total, 5196 images were captured, with frontal faces appeared in 3203 images. Figure 11 shows examples of test images and detected small faces.

Table 2 shows the performance levels of the face detector at different ranges. The average recall rate of 0.9737 proves that the proposed method detects faces robustly, regardless of the range, in normal home environments. In particular, the method presents an excellent recall rate of 0.9009, even for tiny 13×13 pixel-sized faces. It should also be emphasized that the average precision rate of 0.9980 is quite high, which implies that falsely detected faces rarely occurred.

Table 3 presents the performance levels in actual robot environments when both the robot and the human subjects move freely. The results are somewhat worse compared to those shown in Table 2. However, the results presented in Table 2 are from severe environments in which the subjects are continuously walking around, even under poor illumination, ignoring the robots, which were positioned far away. It is clear that these results are very encouraging and that they provide a sufficiently practical level for various robot applications.

We were unable to compare the performance of the proposed method to that of other methods because other approaches cannot detect frontal faces at a long distance while both the camera and the human subjects are moving. However, the experimental

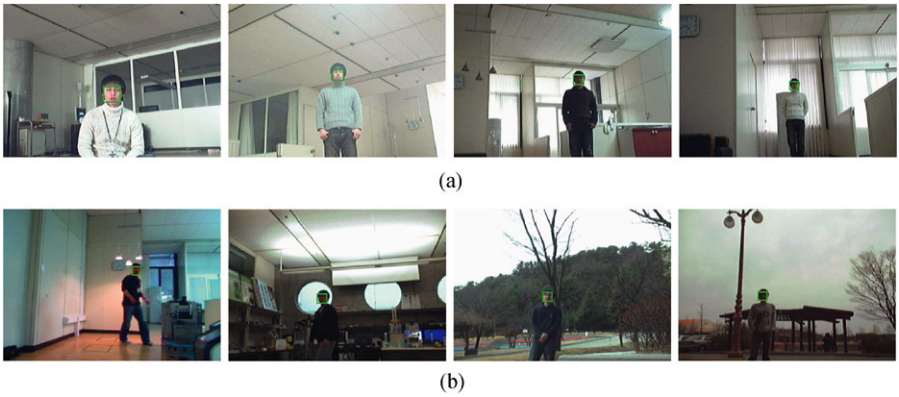

Fig. 11 Examples of detected small faces: (**a**) at different ranges from a database generated in normal home environments; (**b**) from a database generated in actual robot environments

Table 2 The performance of the face detector at difference ranges

Range (m)	Average detected face size (pixels)	N_t	N_c	N_f	Recall rate	Precision rate
1	61×61	4500	4499	0	0.9998	1.0
2	32×32	4500	4500	0	1.0	1.0
3	20×20	4500	4476	4	0.9947	0.9991
4	16×16	4500	4379	32	0.9731	0.9927
5	13×13	4500	4054	8	0.9009	0.9980
Total		22500	21908	44	0.9737	0.9980

Table 3 The performance of the face detector in actual robot environments

Video clip	Description	N_t	N_c	N_f	Recall rate	Precision rate
1	Indoor	357	340	49	0.9524	0.8740
2	Indoor	402	382	23	0.9502	0.9432
3	Indoor	366	331	32	0.9044	0.9118
4	Indoor	487	475	8	0.9754	0.9834
5	Outdoor	467	397	42	0.8501	0.9043
6	Outdoor	450	367	64	0.8156	0.8515
7	Outdoor	329	261	48	0.7933	0.8447
8	Outdoor	345	287	21	0.8319	0.9318
Total		3203	2840	287	0.8842	0.9056

results from the aforementioned large and realistic databases verify that the proposed method can robustly detect tiny faces in robot environments.

5.2 Evaluation of the gesture recognizer

To evaluate the performance of the proposed long-range arm gesture recognizer, 800 video clips of the target gestures and 550 video clips of users' normal actions were collected from ten users within a distance of three to five meters. We captured images using a common USB camera at a speed of 10 fps in a general home environment without constraints on illumination. The size of an image is 320×240 pixels and the type is 24-bit RGB color.

For video clips of target gestures, users were requested to look at the robot and execute arm gestures for approximately three seconds, as shown in Fig. 12. We determined 11 noise gestures, as shown in Fig. 13, in which a human performed acts fre-

Fig. 12 Four target arm gestures

Fig. 13 Determined 11 normal actions by users that occur in everyday life (noise gestures)

Table 4 The performance of the gesture recognizer

Target gestures	Correct (%)	False (%)	Missed (%)	Confusion Matrix (%)				
				Waving	Calling	Raising	Stopping	Noise
Waving	94.50	0.5	5.0	94.50	0.5	0		5.0
Calling	97.96	0	2.04	0	97.96	0	0	2.04
Raising	98.50	0	1.50	0	0	98.50	0	1.50
Stopping	97.98	0	1.52	0	0	0	97.98	1.52
Ave.	97.235	0.5	2.515	–	–	–	–	–

Table 5 Error rate for misclassifying noise gestures as target gestures

	Waving	Calling	Raising	Stopping	Total
Noise gesture	1.85%	2.40%	0%	0.19%	4.44%

quently done in everyday life. Those noise gestures included scratching one's head, clasping one's hands above the head, stretching one's body with raised hands, and other such actions.

Table 4 shows the test results for each target gesture and also presents a confusion matrix.

The average recognition rate was 97.235%, which is a sufficiently practical level for various robot applications. There were no false positive errors apart from one waving gesture being recognized as a calling gesture. "Missed" indicates that the system misclassified the target gestures as a noise gesture, determining that it was a normal everyday action. In this case, the robot may not react to the user's gestures. Although this error is not as serious as a false positive error in robot service, it is certain that this type of error lowers the quality of the service. As shown in Table 4, the proposed system also presents a fairly low error rate.

In addition, we assessed how well the gesture recognizer could differentiate the normal everyday actions from the target gestures. The proposed system is based on spotting-less gesture recognition. That is, the system does not have to detect the start and end position of a gesture. This fact may be a great advantage to the recognizer, but the probability that noise actions are misclassified as target gestures will increase at the same time. In actual robot service, it is necessary for robots to ignore meaningless users' actions.

Table 5 shows the rate of error by the gesture recognizer when it misclassified noise gestures as four target gestures for the 550 collected video clips of users' normal actions. The total error rate was 4.44%. While this is not ideal, it is good enough for the proposed system to be used for actual robot service.

6 Conclusion

This paper introduces a communication method between a robot and users at a long distance. For short-range communication, a speech recognition method is commonly

used. However, at a long distance in which voice signals are comparably weak, a communication method based on gesture recognition can be useful in human–robot interaction.

Therefore, we propose a novel human arm gesture recognition method for mobile robots. It facilitates recognition of four target gestures at a long range of nearly five meters. To detect users, this study presents a novel face detection method that can distinguish tiny frontal faces at a long distance. In particular, the face detector satisfies the requirements of a face detector for mobile robots, in which the subjects' faces should be detectable even when using low-resolution input images captured by moving cameras. The gesture recognizer does not place any constraints on users. Moreover, it is able to distinguish the four target gestures from users' normal actions. We expect that many robot applications will make good use of the proposed method for human–robot interaction at a long distance, where speech information is not generally available.

Acknowledgements This work was supported by the IT R&D program of MKE & KEIT [KI001813, Development of HRI Solutions and Core Chipsets for u-Robot].

References

1. Richarz J, Scheidig A, Martin C, Muller S, Gross H (2007) A monocular pointing pose estimator for gestural instruction of a mobile robot. Int J Adv Robot Syst 4(1):139–150
2. Bremner P, Pipe A, Melhuish C, Fraser M, Subramanian S (2009) Conversational gestures in human–robot interaction. In: IEEE int conf on systems, man, and cybernetics, pp 1645–1649
3. Aggarwal J, Cai Q (1999) Human motion analysis: a review. Comput Vis Image Underst 73(3):428–440
4. Davis JW (2001) Hierarchical motion history images for recognizing human motion. In: IEEE workshop on detection and recognition of events in video, pp 39–46
5. Yin X, Zhu X (2006) Hand posture recognition in gesture-based human–robot interaction. In: IEEE int conf on industrial electronics and applications, pp 1–6
6. Yang H, Park A, Lee S (2007) Gesture spotting and recognition for human–robot interaction. IEEE Trans Robot 23(2):256–270
7. Hwang B, Kim S, Lee S (2006) A full-body gesture database for automatic gesture recognition. In: Int conf on automatic face and gesture recognition, pp 243–248
8. Li H, Greenspan M (2005) Multi-scale gesture recognition from time-varying contours. In: IEEE int conf on computer vision, pp 236–243
9. Bien Z, Do J, Kim J, Stefanov D, Park K (2003) User-friendly interaction/interface control of intelligent home for movement-disabled people. In: Int conf on human–computer interaction
10. Kim S, Lee J, Lee R, Hwang E, Chung M (2008) User-friendly personal photo browsing for mobile devices. ETRI J 30(3):432–440
11. Medioni G, Choi J, Kuo C, Choudhury A, Zhang L, Fidaleo D (2007) Non-cooperative persons identification at a distance with 3d face modeling. In: IEEE int conf on biometrics, pp 1–6
12. Baek K, Jang H, Han Y, Hahn H (2005) Efficient small face detection in surveillance images using major color component and LDA scheme. Lect Notes Comput Sci 3802:285–290
13. Hayashi S, Hasegawa O (2006) A detection techniques for degraded face images. In: IEEE int conf on computer vision and pattern recognition, pp 1506–1512
14. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
15. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: IEEE conf. on computer vision and pattern recognition, pp I-511–I-518
16. Jun B, Kim D (2007) Robust real-time face detection using face certainty map. In: Proceeding of the 2nd international conference on biometrics (ICB 2007), vol 4642, pp 29–38