Home Appliances Control using Hand Gesture Based Commands

Manoj Harsule
Raisoni Group of Institutions

Prashant Ingole
G H Raisoni College of Engineering and management, Amravati

Home Appliances Control using Hand Gesture Based Commands

Manoj D. Harsule[1]
Final year M.E.(Electronics & telecommunication),
G.H. Raisoni College of Engineering and Management, Amravati
Mobile Number:+919665202664
manojharsule@gmail.com

Dr. Prashant V. Ingole[2]
Principal,
G.H. Raisoni College of Engineering and Management, Amravati
prashant.ingole@raisoni.net

Abstract

Gesture refers to expressive movement of human body parts having a particular message to be communicated to a receiver. Gesture recognition refers to understanding meaning of human body part movements, which involves the movements of hand, face, head, arms or body. Human hand gestures are a means of non-verbal interaction among people. They range from simple actions of using our hand to point at and move objects around to the more complex ones that express our feelings and allow us to communicate with others. Gestures have got deep roots in our communication. The remarkable ability of the human is the gesture recognition. On the other hand, the computers have played an incredible role in the development of Human being. Considerable effort has been put towards developing intelligent and natural interfaces between users and computer systems. One long-term attempt in Human-Computer Interaction (HCI) has been made by means of speech recognition and it has been a topic of research for decades. Tremendous progress has been made in speech recognition, and several commercially successful speech interfaces have been deployed. However, it has only been in recent years that there has been an increased interest in trying to introduce other human-to-human communication mode of forms into HCI. This includes a class of techniques based on the movement of the human arm and hand, or hand gestures. First attempts to solve this problem resulted in mechanical devices that directly measure hand and/or arm joint angles and spatial position. This group is best represented by the so-called glove-based devices. Second type includes Non-intrusive Hand Gesture technologies which do not contain any device on the hand. Image processing is used for this type of approach. Our effort is to develop such a system that will communicate with the computer so as to be used to control another system.

1. Introduction

Our designed system in its first stage i.e. Recognition stage, captures the image. Then it processes on the captured image and compare with the database images. Each database image is set to the command interface mode [1]. If a particular image is identified then a command is sent to the microcontroller. In its second stage the microcontroller identifies the command and sends signal to the reference port for operation. The system's conceptual block diagram is shown in the figure 1.

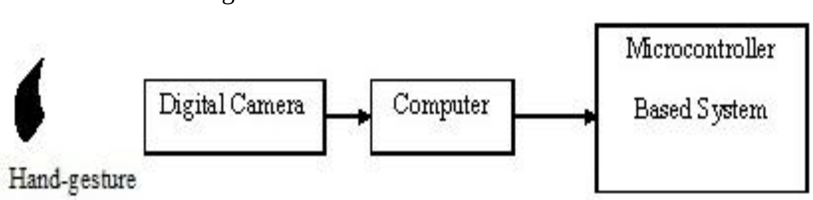

Figure 1. Block Diagram of Human-Machine interaction system

The image recognition can be done with the help of digital camera and software like Microsoft Visual Basic 2008, MATLAB, C#, JAVA etc [1, 3]. Our approach is implemented by using MATLAB. The problem with the MATLAB is that it is a slow tool but our algorithm eliminates the long processing time problem. The main requirement of Hardware is Camera which acts as an input to the computer.

The flowchart for the "Human-Machine interaction using hand gesture based commands" is shown in the figure 2 gives the working of the system. Initially, as shown in figure 2, all the devices are set to the command interface mode. The connected devices to the microcontroller may be either ON/OFF. Once the command interface mode is activated, the computer is ready for the communication with all devices interfaced with the help of microcontroller. Then in next step the camera captures the image and start processing on the captured image. The captured image is compared with the database images for best match. If match of images is detected then computer sends the trigger signal to the microcontroller. The microcontroller converts this command into equivalent ASCII value and commands the device to change its state [2]. If no match is found while processing, there will no state change in the devices state i.e. they will remain as it is (Previous state ON/OFF). After working on one gesture the process of capturing images should not be stopped so images are captured continuously and again the process continues till EXIT operation is applied to the system.

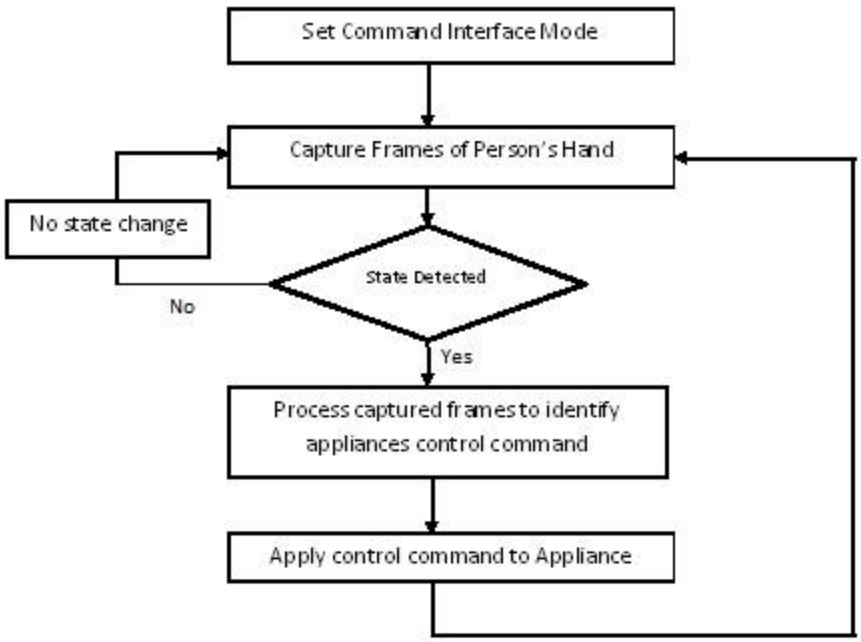

Figure 2. Home Appliances control using hand gesture based commands

2. System components

Operation of our system is mainly divided into two sections. In first stage the digital camera captures the image. It acts as an input to the computer. Captured image is then processed for the match from database images for this we are using the SIFT algorithm. Once the match is found then in the second stage computer generates trigger signal for the microcontroller though serial port. The microcontroller based system may be any robot, a controller in a process control system, another computer or any communication system. In our implementation we have used a switching system for home appliances which changes its state to ON/OFF after getting the signal of hand gesture.

2.1 Hand gesture recognition using SIFT

It is one of the most important operations of the system. The accuracy of the system is totally dependent on this operation. The approach of Scale Invariant Feature Transform (SIFT) for feature detection taken in our implementation is similar with the one taken by Lowe [4, 5], which is used for object recognition. According to his work, the invariant features extracted from images can be used to perform reliable matching between different views of an object or scene. The features have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination[6]. The details of SIFT algorithm is explained as follows.

- **Scale-space peak selection**

The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. Further it has been shown under reasonable assumptions that must be based on the Gaussian function. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image. Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale as presented in figure 3.

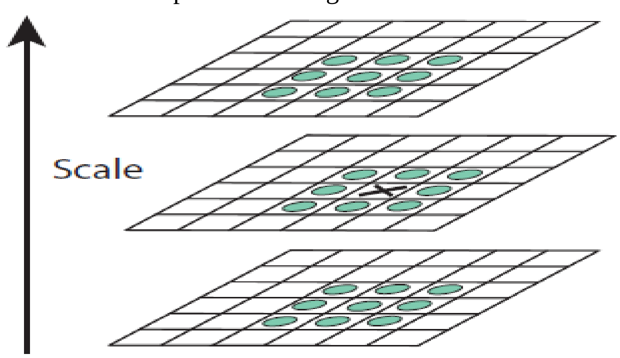

Figure 3. An extrema is defined as any value in the DoG greater than all its neighbors in scale-space

If this value is the minimum or maximum of all these points then this point is an extrema. The search for extrema excludes the first and the last image in each octave because they do not have a scale above and a scale below respectively. To increase the number of extracted features the input image is doubled before it is treated by SIFT algorithm, which results in increase in the computational time [5].

- **Keypoint Localization**

At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability. The local extrema with low contrast and such that correspond to edges are discarded because they are sensitive to noise. The quadratic function is computed using a second order Taylor expansion having the origin at the sample point [6].

- **Orientation Assignment**

One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. For each image sample, L(x, y), at this scale, the gradient magnitude, m(x, y), and orientation, θ (x, y), is precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x+1,y) - L(x-1,y))2 + (L(x,y+1) - L(x,y-1))2} \quad (3)$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y))) \quad (4)$$

The gradient magnitudes are weighted by a Gaussian window whose size depends on the feature octave. To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema [5].

- **Keypoint Descriptor**

These are weighted by a Gaussian window, indicated by the overlaid circle. The figure 4 shows a 2x2 descriptor array computed from an 8x8 set of samples. A keypoint descriptor is created by calculating m(x, y) and θ(x, y) at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region [5, 7].

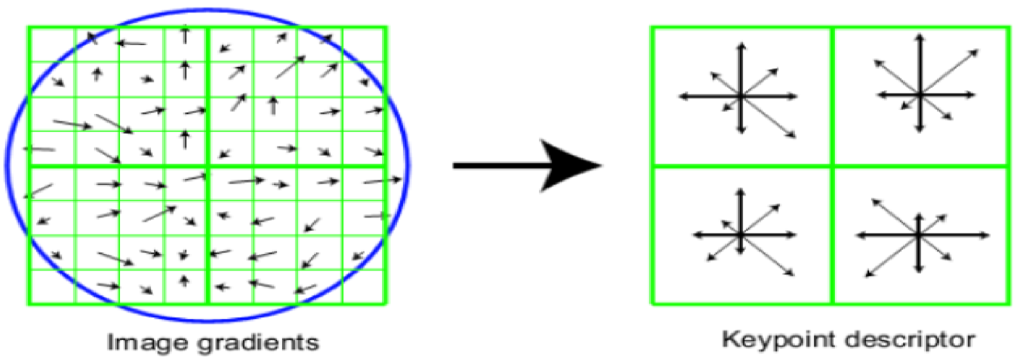

Figure 4. A keypoint descriptor from image gradients

2.2 Command modes

The command interface mode is divided into two sets.

2.2.1 Train set: Some training sets of images, each one is set for one command. Number of users can store their gestures; each one for a device. Once these images are stored for the respective device can be used further anytime by assigning the database path.

2.2.2 Test Set: In the test set the command mode is set-up and camera start capturing the continuous images. When any image is captured, it processed by using SIFT algorithm and it is matched from the database images. The command set in the train set, sets the device ON/OFF with the help of microcontroller.

2.3 Hardware for interfacing

As the computer is connected with the serial port to the microcontroller system, MAX 232 is used for communication between them. Serial port is not available in most of the laptops for that an USB to serial port cable can be used. In our system we have used ATmega 16 microcontroller [2].

3. Result

In our implementation the background is strictly black because shadow does not add noise in the image under the adequate light condition. We have chosen the input images as grayscale so as to save the time of conversion. The result of our implementation is shown in the figure 5. It consist of four figures a to d which will change state of the devices 1-4 as ON/OFF respectively depending on the gesture stored in the database for that image.

Figure 5 (a)

Figure 5 (b)

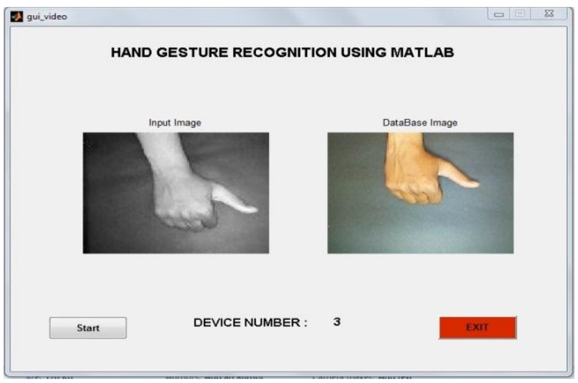

Figure 5 (c)

Figure 5 (d)

Figure 5. Output for different devices (1 to 4) of different gestures

4. Conclusions

With the help of our algorithm we were able to decode gestures successfully and could control the home appliances successfully. The SIFT features described in our implementation have been computed at the edges which are invariant to scaling, rotation and illumination [7]. These features are useful due to their distinctiveness, which enables the correct match for keypoints between different hand gestures in different conditions. The proposed approach was tested on real images. It requires less time for matching in case of grayscale images than color images.

4. References

[1] A. Just," Two-Handed Gestures for Human-Computer Interaction," Ph.D thesis, IDIAP–RR, 6(73), 94, 2006.
[2] http://iitbhu.ac.in/events/ece/aayam2012/wp-content/uploads/2012/03/tutorial.docx.
[3] C. Chang, J. Chen, W. Tai and C. Han, " New Approach for Static Gesture Recognition," Journal of Information Science and Engineering 22, 1047-1057 , 2006.
[4] Lowe D.G., "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision 60(2), 91–110 (2004)
[5] Lowe D.G., "Object recognition from local scale-invariant features," International Conference on Computer Vision (1999)
[6] Brown M. and Lowe D.G. "Invariant features from interest point groups," British Machine Vision Conference, Cardiff, Wales, pp. 656-665.
[7] Lowe D.G., "Local feature view clustering for 3D object recognition," IEEE conference on Computer Vision and Pattern Recognition, Kauai, Hawaii (December 2001), pp. 682-688.