Hand Posture Recognition with Standard Webcam for Natural Interaction

César Osimani[1], Jose A. Piedra-Fernandez[2], Juan Jesus Ojeda-Castelo[2](✉), and Luis Iribarne[2]

[1] Applied Research and Development Center on IT (CIADE-IT),
Universidad Blas Pascal, Córdoba, Argentina
cosimani@ubp.edu.ar

[2] Department of Informatics, Applied Computing Group (ACG),
University of Almeria, Almeria, Spain
jpiedra@ual.es, joc028@inlumine.ual.es, luis.iribarne@ieee.org

Abstract. This paper presents an experimental prototype designed for natural human-computer interaction in an environmental intelligence system. Using computer vision resources, it analyzes the images captured by a webcam to recognize a person's hand movements. There is now a strong trend in interpreting these hand and body movements in general, with computer vision, which is a very attractive field of research. In this study, a mechanism for natural interaction was implemented by analyzing images captured by a webcam based on hand geometry and posture, to show its movements in our model. A camera is installed in such a manner that it can discriminate the movements a person makes using Background Subtraction. Then hands are searched for assisted by segmentation by skin color detection and a series of classifiers. Finally, the geometric characteristics of the hands are extracted to distinguish defined control action positions.

Keywords: Natural interaction · Image processing · Hand gesture recognition

1 Introduction

Among the basic needs in intelligent environments is the supply of personalized information to users through embedded systems in which these users can interact naturally with devices. Therefore, Hand Posture Recognition (HPR) techniques are of interest to facilitate daily life. HPR applications are on the way to being used to control home appliances, for interaction with computer games or for sign language translation. HPR is another input communicating with ubiquitous systems for achieving intuitive and natural interaction. The purpose of this study was detection of hand movements in real time. This is not easy due to the number of variants in forms and viewpoints hands can appear in, showing the palm or fist, partially hidden and with a wide variety of finger positions.

Á. Rocha et al. (eds.), *Recent Advances in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 570, DOI 10.1007/978-3-319-56538-5_17

The real-time detector proposed is based on segmentation by Background Subtraction, face and skin-color detection, supported by edge detection and analysis of geometric shapes.

The main selling point of hand gesture recognition is that you do not need to touch any input device. In human-computer interaction, there are examples of control by several types of hand movements. This paper presents a definition of basic hand-movements and gestures for interaction in a user interface. This system is intended to support user demands in real time. Identifying and following hands requires a robust system that is able to recognize the complex structure of the hand, follow it and interpret it. Some studies on real-life applications are described in [14,23].

The application domains of hand gesture recognition [17] are: desktop applications, sign language, robotics, virtual reality, home automation, smart TV, medical environment, etc.

In the sign language recognition [4], the hand segmentation is a key task in the gesture recognition process, that's why some authors use HSV [5] color model or YCbCr [1]. Currently, there are techniques related to Machine Learning such as Hidden Markov Models [21] or Neural Network [9], which are used in the recognition process. In fact, some projects are integrating devices as Microsoft Kinect [12], Intel RealSense [8] or Leap Motion [15], in order to identify the different signs of this language.

In home automation there is a project called Wisee [16] which is a system that detects when a person is doing a gesture in everywhere at home because the system works through WiFi.

In games, it should be pointed out that educational serious games recognize gestures to improve the learning and physical skills in preschool children [7].

The gestures in the field of medicine are useful to interact with medical instruments, control the resources management of a hospital and special needs people have an alternative way to interact with the computer. In addition, in this case [20] it had the aim of developing an intelligent operating room. This operating room is composed by four subsystems where one of them is hand gesture recognition. The user is able to move ray-X images, select the history of a patient from the database or write down a comment in the image.

The rest of the paper is organized as follows: Sect. 2 describes the methods which has been implemented in order to develop a functional prototype. Section 3 illustrates the prototype developed and the process used to evaluate it. Section 4 summarizes the conclusions and discusses the future work.

2 The HPR Proposal for Natural Interaction

The methods used are described below: Background subtraction, classifier cascade for face, skin colors detection, shape identification and tracking characteristics. The prototype was developed in C++ along with Qt libraries (for the graphical user interface and event management), OpenGL (for 3D effects in presenting processed images), the standard C++ library and Open CV library,

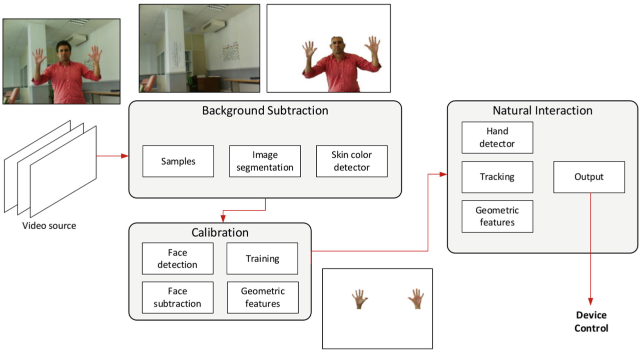


Fig. 1. The schema of the workflow system

which provides a large number of implementations of the algorithms most widely used in image analysis and processing. Figure 1 shows an overview of our work flow and the subsections below discuss each technique used in detail.

2.1 The Recognition Process

The gesture recognition process is organized in three main parts: Background Subtraction, Calibration and Hand Pose Recognition.

The Background Subtraction process separates the background of the image obtained from the camera with the skin color detection method. In this way, the user image is obtained in order to detect hands and the background is ignored.

The calibration process is responsible for identifying the position of the hands. First of all, the face location is detected to make the process faster and more efficient, since the hands will be close to the face and thus the process will not seek in unnecessary regions. Once the face has been identified, it is removed as the background and we get the hands position.

Finally, in the process of HPR, is used the Lucas-Kanade algorithm to track the hands and identify its position at all times. Then, the geometric features extraction method identifies the contour of the hand and a few significant points in the fingers are placed to identify if the hand is close or open.

The following sections explains in more detail the main methods of the process.

2.2 Background Subtraction

Background subtraction is a method used in computer display to separate foreground objects from the background in images captured by a stationary video camera by calculating the differences between frames. This technique saves samples of previous images in the memory and generates a background model based on statistical properties of those samples. From there, a binary image is constructed that acts as a mask for segmentation or separation of background and foreground objects. In addition, a Gaussian blur filter is used to remove brusque changes in images. Figure 1 shows the result of background subtraction.

2.3 Cascade Classifiers

OpenCV implements face detection by a statistical method based on training samples (images with faces and images without faces) from which information is extracted that distinguishes a face from one which is not a face. One of the most widely used methods, developed by Viola and Jones [18], which trains to determine the characteristics of a particular object (face, eyes, hands, etc.), comes from this idea. In this study, class CascadeClassifier (available in OpenCV) is used in face detection for two purposes: To detect the starting position of the hands with respect to the face at the time of system calibration, and to identify image regions where it is unnecessary to search for hands. Starting conditions for correct use of this prototype are defined and one of them is the starting location of the person for beginning the calibration stage.

2.4 Skin Color Detection

Segmentation is complemented with skin-color detection to reduce the hand detection region. Lab Color space is used because it translates a change in color into a change with closely matching visual importance. The three parameters represent luminosity ($L = 0$ black and $L = 100$ is white), its position between red and green (a, green is negative and red is positive) and its position between yellow and blue (b, negative for blue and positive for yellow). Good results are obtained by selecting 109 to 133 for a component.

2.5 Geometric Features Extraction

Hand geometry extraction begins by using the OpenCV findContours function which returns a set of points that form the contours in the binary input image [6]. Some contours may be contained within others. However, only the outermost contours are retained and they are filled in with a solid color. This way, a new binary image is created which works like a mask to obtain a delineated image of the hands. The points on the convex hull of this image are likely to be fingers. However there will also be other convex points because part of the arm may appear in the image. One way to identify the convex points corresponding to finger tips is to make use of convexity defects. The points that make up the convex contour are found by making use of the convexHull function, and the points in the convexity defects are calculated using the convexityDefects function. The convex hull includes the contour of the hand and by selecting the points in each segment of it that are separated the most from this hull, the points forming the bottom of the spaces between the fingers are found. These points are the defects in convexity and enable the number of outstretched fingers to be found very simply. An example is shown in Fig. 2.

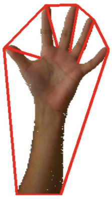


Fig. 2. The convex hull and convexity defects

One advantage of this finger detector is that the algorithm is very simple, and its disadvantage is its low precision in determining the number of extended fingers. However, in this study, it was only necessary to differentiate a closed hand from an open one, and this method is good enough for that. The threshold for distinguishing between a closed hand and an open one is three fingers.

2.6 Tracking Hands

The last stage consists of tracking the hands by implementing the Lucas-Kanade method based on patterns of apparent motion or optical flow. This method, developed by Bruce D. Lucas and Takeo Kanade is a widely used differential method for estimating optical flow [2]. It is assumed that the flow is constant in a local zone of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighborhood using the least frames criterion. It is assumed that movement of the hand in the image between two consecutive frames is slight and approximately constant within the vicinity of the point considered. Then it may be assumed that the optical flow equation is maintained for all the pixels within the vicinity centered on the point considered. Among the OpenCV tools is the goodFeaturesToTrack function which enables a group of pixels good for tracking to be found. As would be expected, a group of pixels good for tracking is one which has texture and edges. This is a problem in images of hands because they have a uniform visual texture. After the groups of points (good features to track) are found, the calcOpticalFlowPyrLK function is used to find the corresponding characteristics or group of pixels in the next frame. Since the result could deliver false positives, the Forward - Backward Error method is used (see Fig. 3). The same method is used in the second frame to estimate the

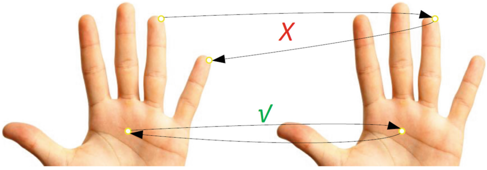


Fig. 3. Forward-Backward Error method

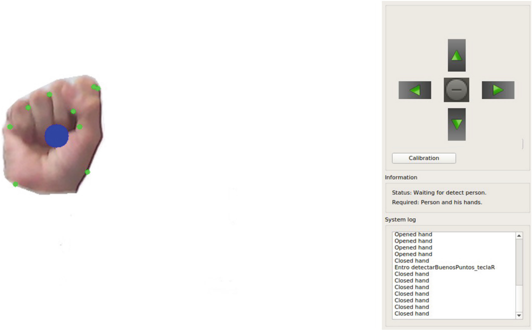
characteristics in the first frame. With this, characteristics common to both are acquired which are the correct tracking characteristics in the second frame.

3 Practice and Experience

This section shows the prototype developed to test the techniques of hand posture recognition described in previous section and the real interface where it has been applied. Finally, it is showed the results about the experiments which have been done in different conditions.

3.1 Prototype Interface

The experimental prototype requires a first adjustment before starting consisting of two steps: capture of images showing background, and exposure of both hands for their calibration. Figure 4 shows the graphical user interface for the first adjustment. When the user presses the Calibration button, a message on the screen tells him to move away from the scene so background images can be taken. Then the user is told to position his body and hands for calibration.



Status: Waiting for detect person.
Required: Person and his hands.

```
Opened hand
Opened hand
Opened hand
Opened hand
Closed hand
Entro detectarBuenosPuntos_teclaR
Closed hand
Closed hand
Closed hand
Closed hand
Closed hand
Closed hand
Closed hand
```

Fig. 4. Graphical user interface of our application

3.2 Application Domain

The concept mashup [22] in web development is related to Web applications which obtained data from other APIs or sources and thus creating a new service which is more useful for the user. The Enia project [19,24] is based on mashups interfaces (see Fig. 5). This kind of interface is the application domain of the gesture recognition flow. The main feature of this project is the developing of a dynamic user interface which is adapted for the user habits. The user interface contains COTS interface components like widgets which are managed by an intelligent agent.

The user can do different actions in the interface such as opening or closing the menu, create a component or move a component. The gestures about opening

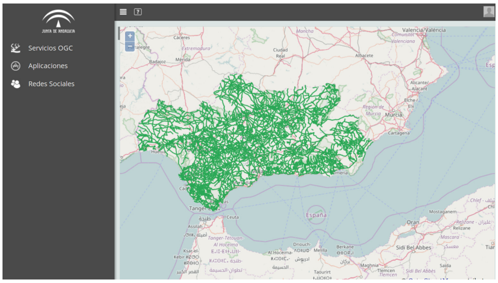


Fig. 5. The ENIA interface

and closing the hand have been included in the prototype of ENIA project and have been associated to the opening or closing menu actions.

The gesture recognition has been designed in order to recognize them regardless of the hand. The aim of this is the interface can be managed by people with hemiplegia. These people just can move a side of their body, for instance the right hand or the left one, thus the system has been designed to recognize gestures with any of the hands. In addition, this feature is really useful for left-handed or right-handed people.

3.3 Experiments

The experimental prototype was evaluated in different environments on an HP ENVY 2.2 GHz Intel Core i7 computer with 8 GB DDR3 SDRAM at a rate of 15 fps with an RGB WebCam with resolution of 640 × 480 pixels. There are no problems in capturing background images as long as the camera remains immobile and does not cause changes in that background. The clearest examples of events that generate problems for background subtraction are changes in scene lighting and shadows a person could cast while using the prototype. Techniques such as those mentioned in [6] can be used to minimize these problems. During the initial calibration, optimum results are obtained in detecting the hand by its geometry and color. Problems that arise during tracking with a decrease in accuracy rates are due to the uniform visual hand texture. This issue impedes detection of good points for tracking. However, the results shown in Table 1 are promising.

Test videos which were included in the test sessions, were recorded with a non-uniform background and three people were doing different movements with one of their hands at the same time. The videos show opening and closing hand motions, and vertical and horizontal movements with the purpose of controlling the interface with joystick. Closing the hand, the users can control the joystick, however, if they open the hand the joystick control is released.

The videos for the test have these features:

- 10 opening and closing hand movements (5 videos with each hand). The posture recognition was tested in two distances: 60 cm and 1 m. The subjects did 200 opening and closing movements and 20 videos were recorded totally.

Table 1. Accuracy rate in hand posture recognition

Motions	Distances			Accuracy
	60 cm	1 m	Two distances	
Opening and closing hand motions	192/200 (96%)	187/200 (93.5%)	379/400 (94.75%)	91%
Horizontal motions with the close hand	183/200 (91.5%)	177/200 (88.5%)	360/400 (90%)	
Vertical motions with the close hand	181/200 (90.5%)	173/200 (86.5%)	354/400 (88.5%)	

– 10 vertical motions with the close hand (5 videos with each hand). The posture recognition was tested in two distances: 60 cm and 1 m. The subjects did 200 opening and closing movements and 20 videos were recorded totally.
– 10 horizontal motions with the close hand (5 videos with each hand). The posture recognition was tested in two distances: 60 cm and 1 m. The subjects did 200 opening and closing movements and 20 videos were recorded totally.

This process was tested with three different people, using a distinct background. The video dataset has 180 videos with 1800 movements totally. The information in Table 1 shows a hit rate of 91%. However, it's necessary to know that the users who made the test, knew the interface operation, therefore, the hit rate wasn't reduced because of inexperienced users.

4 Conclusions and Future Work

This paper has shown evidence that the background subtraction technique, when the camera is kept stationary, is an excellent option for segmenting images. It should also be emphasized that the accuracy rate is increased by combining different techniques to detect the same object, without each technique separately having to have a high accuracy rate. Customization of hand movements and their associated effect on the device controlled is very important in natural interaction systems. And this is logical, since in daily life different people may perform the same task with different movements. The most common examples are people who are right-handed or left-handed and perform their tasks with different hands.

Future work is to recognize more gestures and attach them to another actions of the ENIA project such as moving a component or zoom in and zoom out. Apart from hand gesture recognition, we want to add face recognition intended to include this system in adaptive Web user interfaces as an interaction mechanism for selecting and managing components. For environmental intelligence, this prototype could also facilitate work in interaction and modeling of actions and behaviors in an intelligent building.

Acknowledgments. This work was funded by the EU ERDF and the Spanish Ministry of Economy and Competitiveness (MINECO) under Project TIN2013-41576-R.

This work also received funding from the CEiA3 and CEIMAR consortiums. We thank our colleagues from CIESOL and Solar Energy Resources and Climatology research group (TEP165), who provided data and expertise that greatly assisted the research.

References

1. Adithya, V., Vinod, P.R., Gopalakrishnan, U.: Artificial neural network based method for Indian sign language recognition. In: 2013 IEEE Conference on Information & Communication Technologies (ICT), pp. 1080–1085. IEEE (2013)
2. Bouguet, J.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corp. **5**, 1–10 (2001)
3. Dey, S., Anand, S.: Algorithm For multi-hand finger counting: an easy approach. arXiv preprint (2014). arXiv:1404.2742
4. Ghotkar, A., Kharate, G.: Study of vision based hand gesture recognition using Indian sign language. Computer **55**, 56 (2014)
5. Ghotkar, A., Khatal, R., Khupase, S., Asati, S., Hadap, M.: Hand gesture recognition for Indian sign language. In: 2012 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–4. IEEE (2012)
6. Hasan, M., Mishra, K.: Novel algorithm for multi hand detection and geometric features extraction and recognition. Intl. J. Sci. Eng. Res. **3**, 1–12 (2012)
7. Hsiao, H., Chen, J.: Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills. Comput. Educ. **95**, 151–162 (2016)
8. Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using real-sense. In: 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp. 166–170. IEEE (2015)
9. Ibraheem, N., Khan, R.: Vision based gesture recognition using neural networks approaches: a review. Intl. J. Hum. Comput. Interact. (IJHCI) **3**, 1–14 (2012)
10. Intachak, T., Kaewapichai, W.: Real-time illumination feedback system for adaptive background subtraction working in traffic video monitoring. In: 2011 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), pp. 1–5. IEEE (2011)
11. Kolsch, M., Turk, M.: Analysis of rotational robustness of hand detection with a Viola-Jones detector. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004, pp. 107–110. IEEE (2004)
12. Lang, S., Block, M., Rojas, R.: Sign language recognition using kinect. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012. LNCS (LNAI), vol. 7267, pp. 394–402. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29347-4_46
13. Liu, L., Xing, J., Ai, H., Ruan, X.: Hand posture recognition using finger geometric feature. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 565–568. IEEE (2012)
14. Pang, Y., Ismail, N., Gilbert, P.: A real time vision-based hand gesture interaction. In: Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, pp. 237–242. IEEE (2010)
15. Potter, L., Arauillo, J., Carter, L.: The leap motion controller: a view on sign language. In: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, pp. 175–178. ACM (2013)

16. Pu, Q., Sidhant, G., Shyamnath, G., Shwetak, P.: Whole-home gesture recognition using wireless signals. In: Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, pp. 27–38. ACM (2013)
17. Rautaray, S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. **43**, 1–54 (2015)
18. Rautaray, S., Agrawal, A.: A novel human computer interface based on hand gesture recognition using computer vision techniques. In: Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia, pp. 292–296. ACM (2010)
19. Vallecillos, J., Criado, J., Iribarne, L., Padilla, N.: Dynamic mashup interfaces for information systems using widgets-as-a-service. In: Meersman, R., et al. (eds.) OTM 2014. LNCS, vol. 8842, pp. 438–447. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45550-0_44
20. Wachs, J.: Gaze, posture and gesture recognition to minimize focus shifts for intelligent operating rooms in a collaborative support system. Intl. J. Comput. Commun. Control **V**, 106–124 (2010)
21. Yang, W., Tao, J., Xi, C., Ye, Z.: Sign language recognition system based on weighted hidden Markov model. In 2015 8th International Symposium on Computational Intelligence and Design (ISCID), pp. 449–452. IEEE (2015)
22. Yu, J., Benatallah, B., Casati, F.: Understanding mashup development. IEEE Internet Comput. **12**, 44–52 (2008)
23. Zhu, S., Guo, Z., Ma, L.: Shadow removal with background difference method based on shadow position and edges attributes. EURASIP J. Image Video Process. **2012**, 1–15 (2012)
24. The ENIA Project: http://acg.ual.es/projects/enia/