# Natural Gesture Based Interaction with Virtual Heart in Augmented Reality

Rawia Frikha(✉), Ridha Ejbali, Mourad Zaied, and Chokri Ben Amar

REGIM-Lab: REsearch Groups in Intelligent Machines,
University of Sfax, ENIS, BP 1173, 3038 Sfax, Tunisia
{frikha.rawia.tn,ridha_ejbali,mourad.zaied,ckokri.benamar}@ieee.org

**Abstract.** Augmented reality AR is a relatively new technology that blends digital content such as information, sound, video, graphics, or GPS data into our real world.The natural interaction of users with virtual content is one of the big challenges of the AR application. In this paper, we present a new approach to natural interaction with virtual objects employing users' hands and fingers. We use a real-time image processing system to track gestures from users as a first step and to convert the gestures' shapes on object commands as a second step. The paper describes how these techniques were applied in an interactive AR heart visualization interface. Our aim is to provide an interactive learning tool that will help students to learn about the components of the heart. Experimental results showed the effectiveness of the proposed method.

**Keywords:** Augmented reality · Virtual objects · Hand tracking · Hand gestures recognition · Human computer interaction

## 1 Introduction

AR has become, in a few years, a new mass medium and an emerging technology field. It allows users to view digital content clearly superimposed on the real environment in real time [1,2]. Moreover, there are many possible areas that could benefit from the use of AR technology such as engineering, entertainment and education.

Intuitive interaction techniques are a major aspect of AR applications that allow users to manipulate virtual objects without the aware utilization of prior knowledge. For that purpose, different interaction techniques and concepts have emerged in the field of AR [3]. Vision-based hand gesture recognition is a major driver for intuitive interaction [4,5]. Computer vision algorithms process a video stream, detect the user's hands, and determine a gesture. The gesture launches an interaction function of the AR application that manipulates a virtual object. However, hand gestures recognition is still a challenging research field. In the field of AR, the techniques have not left the search laboratories until today. One reason is the need for technical devices that are attached to the user's hand in order to track it. The user also still acts as an operator of a machine. So,


K. Jackowski et al. (Eds.): IDEAL 2015, LNCS 9375, pp. 457–465, 2015.
DOI: 10.1007/978-3-319-24834-9_53

the interaction is not intuitive. Hand gestures and computer vision-based hand gesture recognition are used to ensure the natural interaction [7]. A user can interact with a virtual object as s/he interacts with a physical object using his/her hands. One advantage is, that a user does not need to wear or carry any technical device in his/her hand.

In this paper, we have developed a set of interaction techniques specifying how the user can manipulate the AR virtual content with free-hand gestures, such as rotation, scaling, etc. By using computer vision algorithms to capture the hand and identify the fingers gestures, we can map the motion of the hand to commands. No additional devices need to be attached to the hands of the user. Our technique aims at improving intuitiveness by supporting natural gesture interaction.

We organize this paper into four parts: first, we place our work in perspective of related work, discussing how hand interaction has previously been implemented in AR environments. Next, we explain the implementation of our approach. Then, we describe the results of our informal usability tests. Finally, we conclude with a summary and an outlook for future works.

## 2 Related Work

There were a number of previous research projects that have explored hand- and finger-based interaction in Augmented Reality interfaces. The most traditional approach for interacting with AR content was the use of vision tracked fiducial markers as interaction devices [6–9]. Piekarski et al. [6] developed a wearable outdoor AR system. They used special gloves for interaction with the system and the environment. Fiducial markers on the user's thumbs are visually tracked by a camera mounted on a head mounted display (HMD). Buchmann et al. [7] developed a technique for natural, fingertip-based interaction with virtual objects in AR environments. Similarly to Piekarski's system, fiducial markers were attached on each finger to track gestures from the user. These capabilities are demonstrated in an application of AR urban planning. However, these two works used markers and the fingers looked awkward with the markers attachments and they didn't have to occlude each other in order to be detected. Other works, such as [8,9] partially addressed encumbrance issues due to multi-marker use by attaching color stickers on fingertips. Hurst et al. [8] presented various interaction techniques on mobile phones by using the system's sensor data, and color markers on fingertips. They used single fingers and couples of fingers for interactions. Thus, the user can apply some operations such as translation, rotation, scaling on the virtual object based on finger gestures. Mistry et al. [9] developed a wearable gestural interface that tracks the location of coloured markers or visual tracking fiducial on user's fingers and projects visual information on a surface. Reifinger et al. presented also a similar system [10] by replacing fiducial markers at the fingertips with optical sensors. They used an infrared hand tracking system and gesture recognition. The tracking system detects the markers and a computer-internal hand model is built using this data. Thus, the user becomes able to manipulate virtual objects as real objects.

The accuracies in gesture detection by marker-based or optical-based inputs are high, however, privacy devices, for example, optical sensors and data gloves are required and their attachments to the fingers limit the freedom of hand movements. In order to keep the AR set-up costs low and to avoid the hand movement limitations, computer-vision based hand gesture recognition was suggested as an alternative.

## 3 Approach

In this paper, we propose a new approach to natural interaction with virtual objects using hand gestures in AR environments. Our proposed approach uses a real-time image processing system to interpret gestures. The flowchart of our prototype system consists of three main modules (Fig. 1).

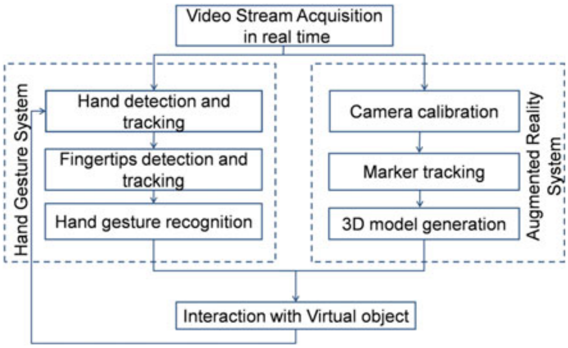


**Fig. 1.** The overall scheme of the 3D virtual objects' command.

- AR system: This module is responsible for connecting and capturing images from the camera and then processing this output to track the marker. The output of this module is a 3D model superimposed on the real world.
- Hand gesture system: This module is responsible for the detection and the tracking of hands and fingers. The output of this module is the hand gesture.
- Interaction module: This module is responsible for mapping hand gestures into functional input and interacting with the virtual heart.

### 3.1 AR System

In this work, we used a monitor-based AR system. The hardware required is a webcam for image acquisition, computing equipment to process the video and a monitor for display. In AR research, marker-based camera pose estimation approaches [11] have shown effective recording of virtual objects with the aid of robust detection of fiducial markers. For this reason, we use ARToolKit, an open source marker based on a tracking software. The operating principle of this software is to calculate the basic geometric transformations to position 3D objects in a real-time image. First, it operates by searching black and white planar markers for each frame of the image. Second, it calculates the position and orientation of the marker relative to the camera. Finally, the computer generates 3D objects and superimposes it on the marker. The first and the important setup of an AR application is the camera calibration that is used for alignment of the marker and the object to obtain correctly the combination of the real and the virtual worlds. So, a camera calibration application included with the ARToolKit library is used.

For rendering, we used OpenVRML (www.openvrml.org), an open-source parser and renderer for VRML97 and X3D files, including support for texturing, animation and networked content. The output of this system is an augmented 3D objects that the user can observe on the screen of the monitor.

### 3.2 Real-Time Hand Tracking

In order to command a virtual object by hand gesture, it is necessary to detect the hand in a stream video first. So, to accomplish this task, we applied the skin color detection algorithm on each input image. Several researches have shown that the main variance is in intensity rather than chrominance. Generally, HSV and YCbCr color spaces help to retrieve from the intensity variations [12]. In our work, we used HSV color space. A big advantage of this color space is that it is less sensitive to shadow and uneven lighting. Generally, the acquired images are presented in the RGB (red, green, blue) color space. For that reason, we convert the input image to HSV color space. Then, a threshold is applied to all the pixels of the image to detect the skin areas. In our implementation, we used the value suggested by Y. Wang [13] for the discrimination of skin and non-skin pixel. It's defined by this equation:

$$\begin{aligned}&((H >= 0)\ \mathit{and}\ (H <= 50))\\&\mathit{and}\ ((S >= 0.20)\ \mathit{and}\ (S <= 0.68)\mathit{and}\ ((V >= 0.35)\ \mathit{and}\ (S <= 1.0))\end{aligned} \tag{1}$$

Next, filter median and smoothing are applied on the segmented image to eliminate the small noise regions. The output of this step is a binary image whose skin areas are represented by the white color and the other non-skin areas are represented by the black color.

After skin regions are detected, we must determine which regions correspond to the hand. For this purpose, we first extract the contours of all skin regions detected in the binary image using contour detection operations. For each region $R_i$, we obtain a set of coordinates $C_{i(j)} = (x_j, y_j)$ of the perimeter which outlines each region. $N_i$ represents the total number of perimeter coordinates in the contour $C_i$. We then select the most significant outline to represent the outline of the hand, using Ni as a measure of the contour size. $N_i$ must exceed a threshold S to ensure that the contour $C_i$ is considered (S = 55 in our case).

Once the hand is detected, it must be tracking. The idea is to use a tracking method based on the detection of points of interests. So, the idea is to look for areas of the image where there is a strong contrast change, and this is particularly the case with the points lying on the edges. We will concern ourselves with the track corners, areas where there is a strong gradient in two orthogonal directions. We use the cvGoodFeaturesToTrack() method, implemented in the OpenCV library.

In the tracking phase, simple images that do not have multiple hands do not cause problems. However, the tracking problems worsen with the increase of the hands detected in the image and especially in the case of overlapping faces and hand tracking, they can give false results.

### 3.3 Real-Time Finger Tracking

In order to detect the fingertips, we use the k-curvature algorithm [14] which is used to find pixels that represent peaks along the contour perimeters of hands detected. First, at each pixel i in a hand contour, we compute the angle $\alpha$ between the two vectors of three points $[C(i), C(i-k)] and [C(i), C(i+k)]$ using the following equation where k is a constant that was fixed after some tests:

$$\alpha_{C_i} = \arccos \frac{a^2 + b^2 - c^2}{2ab} \tag{2}$$

where a is the distance between $[C(i), C(i-k)]$, b is the distance between $[C(i), C(i+k)]$, c is the distance between $[C(i-k), C(i+k)]$ and $\alpha_{C_i}$ is the angle between $[C(i), C(i-k)]$ and $[C(i), C(i+k)]$. The idea here is to find points that represent potential peaks or valleys along the perimeter as shown in Fig. 2.

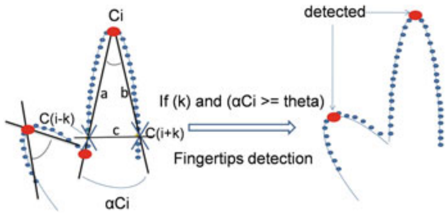


**Fig. 2.** Illustration of the k-curvature

To evaluate the stage of fingertips detection, a base of 100 test images is created. For this test we have chosen images containing: one hand, two hands and more than two hands in several forms, under different lighting conditions and from different skin colors. The evaluation protocol at this phase consists in testing the 100 test images by changing the value of k in the range [10..30] and by varying $\alpha$ in the range [30 %–90 %]. This stage has as goal to determine the rate of correct detections of the fingertips based on the two variables $\alpha$ and k. The good detection rates are calculated as follows:

$$Good\ detection\ rate = \frac{number\ of\ detected\ fingers}{total\ number\ of\ fingers\ in\ images} \tag{3}$$

The results of applying the evaluation protocol used in the detection will be displayed in the Fig. 3 below. According to the figure above, we notice that the rate of good detection depend on the two variables $\alpha$ and k. So, by varying $\alpha$ and k, we obtained a good detection rate (89 %) with k set to 30 and $\alpha$ between 70 % and 90 %. Figure 4 shows the results of fingertips detection, the fingertip is marked with a blue point. Our method can detect the fingertips under different lighting conditions (Fig. 4(a) and 4(b)) and background clutter (Fig. 4(c)).

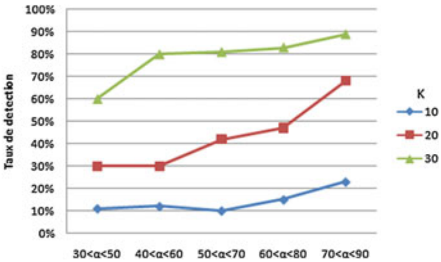


**Fig. 3.** Variation in the rate of good detections based on the values of K and $\alpha$

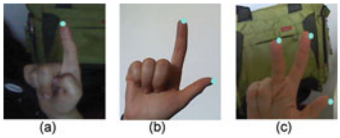


**Fig. 4.** Fingertips detection results using k-curvature. (a) Dark situation. (b) Natural lighting condition with white background. (c) Natural lighting condition with cluttered background (Color figure online).

### 3.4 Interaction Module

This section describes a set of interaction techniques specifying how the user can manipulate the AR virtual heart. All interactions are performed by free-hand gestures without any devices attached to the user's interaction. The proposed system is tested using an Intel $CORE^{TM}$ i3 CPU with a 4-GO of RAM and the windows 7 operating system. The test image sequences were grabbed from a Web camera (labtec webcam) with a resolution of 640 by 480. The camera stands opposite to the user. It flows the user's hand in two dimensions and according to the shape of the hands the interaction is made. We used five gestures in our system. Figure 5 shows the command of a 3D virtual heart via hand gesture recognition. When the three, two, one and zero fingers are recognized, the model is rotated, enlarged, attenuated and stopped interaction, respectively. In addition, we displayed heart's components when the four finger are presented. We notice that the result of the command of the AR virtual heart is excellently related to the fingertips recognition performances. The average processing time per frame, which includes preprocessing, rendering of the AR virtual heart, fingertips detection and tracking, and recognition of gestures, is 18.35 ms. This processing time (less than 20 ms) is fast enough for real-time application.

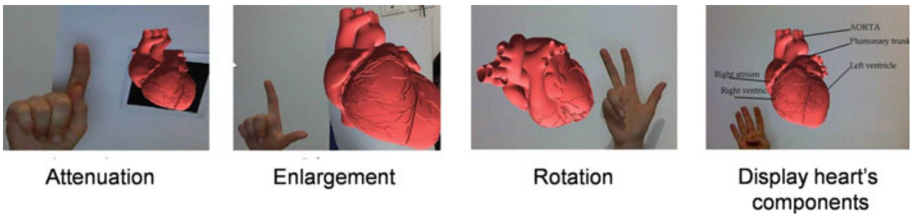


**Fig. 5.** Command of a virtual heart via hand gestures

Our system has been tested by a group of users. The aim of the test was to explore whether the interaction techniques facilitate the manipulation of virtual heart or not. We also want to know if our system complies to user expectations. A group of 15 users participated in the study. We noted that no user has a previous experience with hand gesture based interaction. Participants are between 20 and 55 years old. Before starting the test, the various interaction techniques were presented. Each user has taken few minutes to practise the technique of interaction. During the test, the user may choose which interaction they want to apply in the heart. The user can choose among five types of interaction. He/She must present their hands in front of the camera and use one of five gestures proposed by our system. So, according to the introduced gesture, interaction is made with the heart.

In general most users had no problem using gestures to manipulate virtual object even when they had no a previous experience with AR interaction. In addition, most users were able to interact in an intuitive way with the virtual heart and told us that the interaction was easy and we could manipulate virtual object in the same way as real objects. The results of user test indicated that the use of AR were helpful and facilitate learning components of the heart. Furthermore, our approach not only provides more natural and intuitive interaction but also offers an economical and convenient way of interaction.

## 4 Conclusion

In this work, we presented a learning tool to controlling the virtual heart via natural hand interaction without grasble devices. For this purpose, we use skin color segmentation to detect the hand region. Then, the k-curvature algorithm is implemented to find fingertips and to use it to define the hand gesture. So, a user can interact with the virtual heart by using these techniques. The future work has two objectives. First, we will create a computerized model of a heart and orient it with a hand gesture. Until now, a simple 3D model of heart has been only used. The purpose of this model is to provide a means of visualization and guidance to surgeons during surgery and to facilitate the learning and the work of surgeons. In the next step, the surgeons should be able to simulate the different hand-made gestures during surgery.

**Acknowledgements.** The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

## References

1. Azuma, R.: A survey of augmented reality. Presence Teleoperators Virtual Env. **6**, 355–385 (1997)
2. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Comput. Graph. Appl. **21**(6), 34–47 (2001)
3. Zhou, F., Been-Lirn Duh, H., Billinghurst, M.: Trends in augmented reality tracking, interaction and display: a review of ten years of ISMAR. In: Proceedings of the 7th IEEE/ACM ISMAR, pp. 193–202 (2008)
4. Radkowski, R., Stritzke, C.: Interactive hand gesture-based assembly for augmented reality applications. In: Proceedings of the 2012 International Conference on Advances in Computer-Human Interactions, pp. 303–308 (2012)
5. Ejbali, R., Zaied, M., Ben Amar, C.H.: A computer control system using a virtual keyboard. In: ICMV (2015)
6. Piekarski, W., Thomas, B.H.: Using ARToolKit for 3D hand position tracking in mobile outdoor environments. In: Proceedings of 1st International Augmented Reality Toolkit Workshop (2003)
7. Buchmann, V., Violich, S., Billinghurst, M., Cockburn, A.: FingARtips: gesture based direct manipulation in augmented reality. In: Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE 2004, pp. 212–221. ACM, New York (2004)
8. Hurst, W., van Wezel, C.: Gesture-based interaction via finger tracking for mobile augmented reality. Multimedia Tools Appl. **62**, 1–26 (2012)
9. Mistry, P., Maes, P., Chang, L.: WUW - Wear ur World: a wearable gestural interface. In: CHI 2009 Extended Abstracts on Human Factors in Computing Systems, pp. 4111–4116. ACM (2009)
10. Reifinger, S., Wallhoff, F., Ablassmeier, M., Poitschke, T., Rigoll, G.: Static and dynamic hand-gesture recognition for augmented reality applications. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4552, pp. 728–737. Springer, Heidelberg (2007)

11. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd International Workshop on Augmented Reality, IWAR 1999 (1999)
12. Chitra, S., Balakrishnan, G.: Comparative study for two color spaces HSCbCr and YCbCr in skin color detection. Appl. Math. Sci. **6**, 4229–4238 (2012)
13. Wang, Y., Yuan, B.: A novel approach for human face detection from color images under complex background. Pattern Recogn. **34**(10), 1983–1992 (2001)
14. Segen, J., Kumar, S.: Fast and accurate 3D gesture recognition interface. IEEE International Conference on Pattern Recognition (1998)