Interaction Control Based on Vision for AR Interface of Smart Phone

Thomas Furness[1] and Young Jae Lee[2*]

[1]*Dept. of Industrial & System Engineering, University of Washington*
Seattle, WA 98195-2650, USA
[2*] *Dept. of Multimedia, Jeonju University*
#45, Backma-Gil,Wansan-Gu,Jeonju,Jeonbuk,560-759, Korea
[1]*tfurness@uw.edu*, [2]*leeyj@jj.ac.kr*

Abstract

This paper proposes methods for detecting and tracking moving body parts such as the eyes, lips, and hands to enable natural interaction with smart phone platforms. For demonstrating these methods we have implemented a markerless-based augmented reality on an Android-based smart phone. For interaction the user does not use any extra appliances except for the natural movement of the eyes, lips and hands within the instantaneous field of view of the smart phone camera. To realize augmented reality, we developed state-of-the art computer vision algorithms to first detect and track the face using an Android API operating on the camera images, then we track the eye location within the face. We estimate the location of the lips rapidly by making inferences based on the information from the eye and face locations. Real-time detection of optimal locations of the movable bodily parts (such as hands) also use the smart phone camera.

These methods allow the user to interact with virtual objects that are displayed on the smart phone. The demonstrated methods address current problems involving delayed detection time and unreliable tracking in markerless-based augmented reality applications. Our suggestions may serve to advance interaction control in smart phone augmented reality and serve as a platform for interface development.

Keywords: *vision-based interaction, AR, interface, smart phone*

1. Introduction

Widespread smart phone use has increased the interest and research in technologies related to augmented reality(AR).[1-17] Current research in AR is actively investigating vision-based interaction and interface in the areas of entertainment, education, medicine, engineering.[1-4, 7-17] Historically AR has been derived from virtual reality, meaning the kind of technology which combines computer-processed virtual information with real-world information. Rather than being a substitute for the real world as in the case of virtual reality (VR), AR adds virtual information to real world objects or scenes there by 'augmenting' the real world, thereby combining related information.

With AR information could be serviced at a lower cost than through virtual reality since it is not necessary to construct an alternative virtual world…hence the virtual information is actually a representation of the real world. Another benefit of the augmented reality is[1] that it can provide interaction in a more familiar sense of reality. AR on a mobile phone is different from the integration of the real-world image information with three-dimensional virtual objects. In fact, it uses more diverse recognition technologies through mobile sensors, and needs to be defined separately as providing augmented information service through augmented interface and service

---

[2*] A corresponding author

mashup.[1-2] Mobile augmented reality is also developing quickly due to the widespread use of mobile phones or smart phones with camera and GPS sensors. Also contributing are fast mobile internet services and location-based related functions. The earliest augmented reality service was focused on displaying virtual objects effectively. On the other hand, recent mobile augmented reality services are more interested in effectively browsing various internet services and information, or in more convenient use of the augmented interface through mobile devices using sensor information and natural interaction [1-4].

Generally augmented reality technologies are divided into marker-based and markerless. In marker-based augmented reality, the location and orientation of the viewing window of the display is estimated in six degrees of freedom relative to a tangible card marker.

Virtual objects can then be positioned dynamically relative to the pose of the marker within the window so that the virtual objects appear to be 'connected' to the real space or objects. Modest computational overhead is necessary to estimate pose and recognize various identifiers in the card marker thereby making this appropriate suitable for real-time use. But the marker approach has its drawbacks in that a particular maker should be present at all times, and the related marker must be 'trained' for augmentation of various objects. Alternatively, markerless AR does not require any particular markers: the characteristics of the input images (*e.g.,* often the real world) are analyzed to create coordinates for the locations of the objects to be augmented. While markerless AR is more natural in interacting with the real world, it does require a substantial increase in computation complexity and is more difficult to realize. [3] This paper proposes a vision-based markerless AR approach using a smartphone platform. The proposed methods are applied for natural interaction of virtual objects presented on the smartphone. For the purpose of the study, we constructed a new smart phone environment for markerless augmented reality using the android smart phone platform. We implemented detection and tracking methods for the face, eyes, and lips enabling natural interaction.

We detect and analyze the optimal areas of the moving parts (hands and *etc.*,) in real-time, thereby suggesting new more efficient interaction methods with objects in the augmented reality applications. The organization of this paper is as follows. In Section 2 we explain the composition of the proposed system. In Section 3 the proposed algorithm is explained and its mechanism is introduced. In Section 4, we propose the area detection methods for moving objects such as face, eye, and lip. Experiments validate the effectiveness of our approach. Section 5 covers the conclusion.

2. Detection of Face, Eye and Lip

Many people have been interested in 'face image processing'.[4-17] Various approaches have been tried for face detection and face recognition notably in security systems, head gesture recognition, and human-robot systems. Efficient non-intrusive face detection and face recognition systems are ideal in that they don't require uncomfortable processing and extra equipment. Detecting face area and extracting characteristic elements from the face are basic and crucial parts of the face image processing. But still there are significant challenges in detecting the image of human face precisely and in real time. [11-16] Faces are especially difficult to detect since they contain factors of face shape, size, image quality, color, orientation, complex background settings and lighting. [13-16] this multiplicity of parameters necessitates complex computational steps including transformation, filtering, and labeling for skin area segmentation.

Our approach does not attempt to detect the face using traditional methods. We use an API on Android to detect face in real time then use that information to locate then track the eyes and lips to enable interaction within mobile augmented reality on the smartphone. Once we have detected the face area, we can extract information on eye, nose, and lips.

Eyes and lips, which are characterizing elements of the face, lie at certain points. The distance between the eyes is also calculable. The first conventional method to detect the eye is using ‘Eye Map’ was proposed by Rein-Lien Hsu, Mohamed Abdel-Mottaleb and Anil K. Jain [3].

In this regard there are two ways to make ‘Eye Map’, one of which is ‘EyeMapC’ which uses the color value. The color value is calculated in the YCbCr color space. The formula for EyeMapC is as follows.[3-4][16]

$$EyeMapC = \frac{1}{3}\{(C_b^2) + (\tilde{C}_r)^2 + (C_b / C_r)\} \qquad (1)$$

$(C_b^2), (\tilde{C}_r)^2, (C_b / C_r)$ are the normalized values from 0 to 255. The value of C_b and C_r respectively, which was inversed from YC_bC_r. The formula is based on the fact that in the skin area, C_b is usually higher than C_r in value. Another way is EyeMapL which uses the contrast effect of light and darkness. The formula (2) is presented here.

$$EyeMapL = \frac{Y(x,y) \oplus g_\sigma(x,y)}{Y(x,y) \ominus g_\sigma(x,y) + 1} \qquad (2)$$

Use the black-and-white image for contrast effect through ‘erode and dilate’ to get ‘EyeMapL’. $\oplus$ means ‘dilate’, and $\ominus$ means ‘erode’. g(x, y) is the contrast value of the black-and-white image. With the two given values, we can get ‘EyeMap’ as in the formula (3).

$$EyeMap = (EyeMap\,C)\,and\,(EyeMap\,L) \qquad (3)$$

Get the candidate area for the eye through binarization of ‘EyeMap’ using threshold value. This method, however, has a problem in that we may not expect to get the eye area only. Other results might come out as well, which makes it unfit for real-time interaction since various calculations cause time delay.

The second method is to use API on Android [7-9] By using ‘FaceDetector’ and ‘FaceDetector.Face class’, we can identify the bitmap size of the input image along with locations and the number of faces present in the image. We can also get CONFIDENCE_THRESHOLD constant information (set at 0.4) along with face Euler values for x–axis, y-axis, and z-axis of face rotations. Based on the information, we can identify the location of the eyes. This method involves less calculation and is relatively secure. So it is good for the AR(augmented reality) environment in which real-time interactions and stable conditions are required. But the detection of the exact location is only possible when the eyes lie on the identical location on the y-axis.

The third method is the proposed way in this paper, and uses the difference image and the smoothing method. To set up the object area, first transform the input image into the grey level and perform the difference image. For fast processing in the difference image, we recommend binarization by setting threshold value for the constant rather than using the usual image of average brightness. Considering the characteristics of the user’s eyes constantly blinking in AR environment, we can capture the eyes with location information in the difference images between frames, and difference in information volume of image. f1(x, y) is the present frame, and f2(x, y) is the previous frame. And to get the difference image d(x, y) between the two frames, we calculate the difference between each pixel pair corresponding to f1 and f2. [5-6]

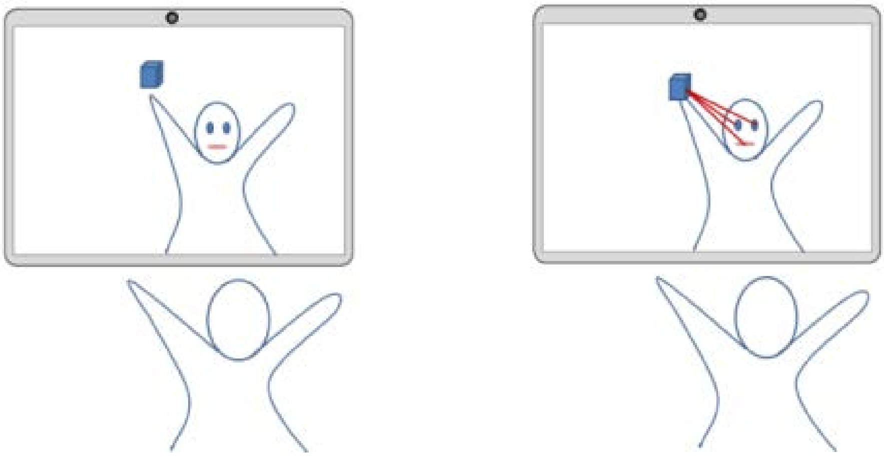
$$d(x,y) = f1(x,y) - f2(x,y) \qquad (4)$$

To reinforce the image information, mean filtering is used to detect the location of the eyes which are characterized by blinking. Mean filtering[6] is a simple, intuitive and easy to implement method of *smoothing* images, *i.e.*, reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images. The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. We used a 3×3 square kernel although larger kernels can be used for more severe smoothing.

This paper opts for the second and the third methods which enable us to recognize the face fast and effectively and also allow for the extraction of the eye information in the face area. We also propose the methods for lip detection through the chosen methods. The results will be used to propose the AR interaction algorithm and its performance is to be verified through experiments.

3. Proposed System Configuration

By using the camera on smart phone and virtual objects, we build the AR environment. Vision-based interaction control is realized between the user and the virtual 3D objects. The selected interface is the detected information on user's face, lip, hand, and arm. (Refer to Figure 1. (a)(b)).



(a) (b)

Figure 1. The Design of System Configuration

Figure (a) in Figure 1 shows the augmented virtual object in the augmented reality environment. Figure (b) displays the interaction with the user; the user interacts with the object which is detected to realize the action and reaction depending on each occasion. In Figure (b), the hand collides with the augmented object at which point the eye and the lip emit energy rays which explode the virtual object. For this effect, the input camera images are used for face detection, and the eye and lip locations are detected by analyzing and inferring based on the face detection information. We infer the movement information of the hand and the arm needed for attacking the virtual object from the difference image and the smoothing image. We also realize natural interaction control with the augmented virtual object. Mobile augmented reality services available so far could be classified, based on technologies of object tracking, into pseudo augmented reality, marker based augmented reality service, and vision based augmented reality service.[6-9] First, pseudo augmented reality service uses sensors applicable on mobile gadgets to identify augmented location on screen and visualize related content. Information is augmented with GPS and digital compass installed on mobile gadgets. Easy realization is possible even without the need for complicated technology, but the drawback is that we cannot expect precise information augmentation on a particular location. Second, marker based augmented reality service utilizes visual markers attached to the object to identify and locate the object. Related information is augmented to provide augmented reality service. The advantage is that you can easily identify the object only with the image from the camera without need for a compass or GPS sensor. But the markers attached to the object are sensitive to light changes and irritate one's eyes. Third, vision-based augmented reality service captures main characteristics from the camera images and identifies and tracks the object to augment related content. The setback is that it requires abundant resources because of accompanying real time image processing and content augmentation. But the advantage is that it is highly mobile. This study is based on the third method, the vision based augmented reality service, which is not bound in space and highly mobile.

4. Proposed Algorithm

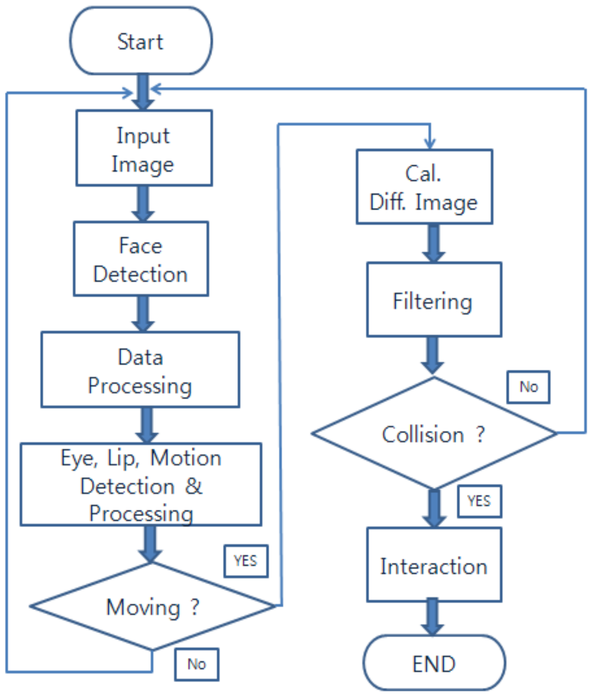


Figure 2. Flowchart of the Proposed Methods

Figure 2 shows the flowchart of the proposed methods. When the image is input on the smart phone camera, we can detect the face through API on Android. Then we can detect the location of the eyes by using the distance between the eyes which is on the detected face area. For this purpose, two methods are used here. The first involves face detection and uses the face detection variables. The second is to make the input image into the grey image and to use the difference image referring to the difference from the previous frame. To reinforce the image information, smoothing is used to detect the location of the eyes which are characterized by blinking (Figure 2 Detection of the eye location through eye detection processing in filtering realization). Also, the ratio information gotten from the experiment is used to enlarge the face area, which is then used to infer and detect the lip area. The interface which only uses the detection of the eye and the lip is not sufficient for realizing augmented reality. So there is a need for additional interface of bodily parts(such as hand) which move naturally. This additional movement information is also to be gotten from the grey image from which the eye detection was done. To control natural interaction, we create two virtual spacecrafts and a cube whose movements collide with body parts. When collision happens in the virtual space, action and reaction interaction is realized by using collision sound and explosion event. When there is no collision, we return to the early routine. The image data used at this point is maintained 'integer data' and processed for maintaining effective compatibility. SAMSUNG Galaxy Tab 10.1 model(Os ver. 4.0)l is used for development under 'Eclipse' environment.

5. Detection Method

5.1. Lip Detection

5.1.1. Method 1:

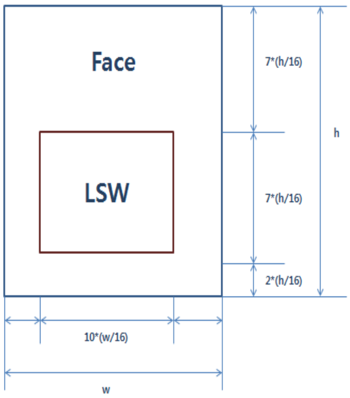


Figure 3. Face and LSW(Lip search window)

Lip area which satisfies the characteristics of the face area and is present within the face statistically can be defined as in Figure 3.[10] LSW(Lip search window) is defined as the area which corresponds to 7*(h/16) from the top in the face height(h), and to 10*(w/16) from both sides relative to center of the face width(w). We search for the lip area according to the definition and through thresholding and outline detection processing.

But this method has drawbacks since it requires complicated image processing involving making outlines by enlarging the oval. It also requires complex process for image transformation. Therefore, this method is not suitable for process and recognition algorithm in the AR environment needing real-time interaction.

5.1.2. Proposed Method:

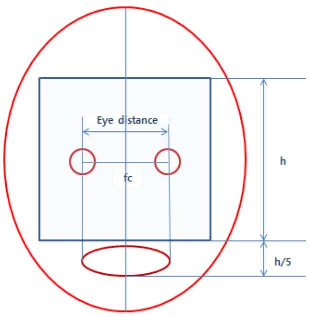


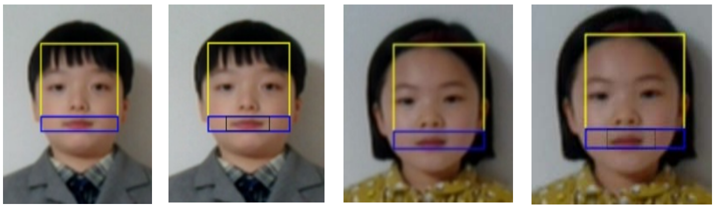
Figure 4. Proposed Methods for Lip Detection (fc: face midpoint h: face height)

Lip is detected by using the face detection information on android API. Face area is detected based on face midpoint(fc, Figure 4). The eye location is detected by using the eye distance. Use the distance between the eyes(Figure 4. Eye distance) and detect the location of the eyes. The eye distance is used to get the height of the face and the distance between the eyes, which is then used to get the size of the lip. (the oval in the bottom of Figure 4) In other words, we set the square for the lip area at 1/5*h of the face height. Since the lip is located in the bottom of the face, we can realize the lip area detection by attaching it to the face recognition area, and eye distance information is used to detect the lip. We used 75 face pictures and applied the proposed method and the detection was possible at 94.7%.

However, when the face was slanted or the eye area was unclear making it hard to detect the face, lip detection was impossible. Though it is lack in precision, the proposed method is suitable for real-time lip detection in AR environment because it is fast reusing the face data.

6. Experiment

6.1. Experiment 1

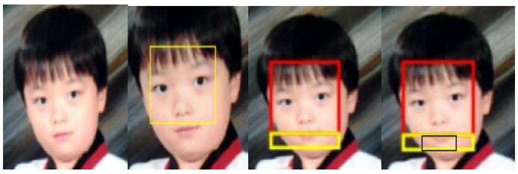


(a) (b) (c) (d)

Figure 5. Resulting Image of Experiment 1

Figure (a)(c) in Figure 5 show the face area enlarged with the proposed method. Figure (b)(d) show the image of lip area detected by using the proposed method in the enlarged face area. Though Figure (b)(d), we can see that the lip area has been successfully detected.

6.2. Experiment 2



(a) (b) (c) (d)

Figure 6. Input Image and the Resulting Image

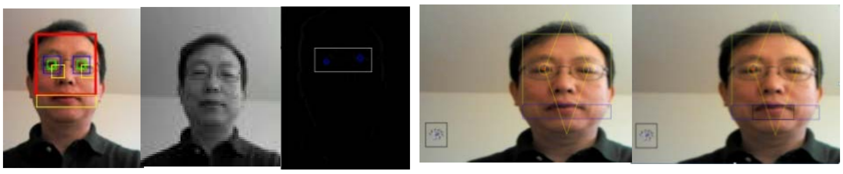
Experiment 2 tries to detect the lip area in the slanted face, and is different from Experiment 1. Figure(a) shows the input image, and Figure(b) is the face detection, and Figure(c) is the face enlarged with the proposed method. Figure(d) deals with the resulting image of the lip area which has been detected by applying the proposed method to the enlarged image. If face detection is possible in the slanted face, lip area is also possible to be detected by proposed enlargement method.

6.3. Experiment 3

In experiment 3, we applied the proposed method to input real-time moving image. 3D virtual object was augmented to experiment construction of vision-based AR environment, and interface for interaction.

We detected face in real time by using moving image input on android smart phone camera and android API. Here we determined the size and location of the face detection area by using the distance between the eyes at face midpoint. (Figure(a) red square) To detect the eye area, two methods are used here which are fast and can provide interface for interaction in the AR environment. In the green square in Figure(a), the eye location has been determined by dividing the eye distance from the face midpoint by two. This method is relatively simple, and can be used to detect the eye only with the location information of face detection. The blue and orange squares in Figure(a) are the eye detection area squares which are to be used to detect the changed pixel data volume around the midpoint gotten from the first method. The blue square and the orange square are the output images to detect the changed value at 15 pixel and at 10 pixel area respectively. Figure (d) is the output image of the center in red and blue. It is limited to the case where the value is greater than the threshold value. We can identify the location of the eye depending on the blinking information. For the left eye of the user, the center of the detected eye area on the difference image is in red circle and the second method was used. In Figure(b), the color image has been changed into the grey image to make the difference image. Figure(c) is the result of detecting the surrounding area of the eyes by using the difference image from the previous frame. Figure(d) shows the lip area in which virtual object was augmented and the face area was enlarged by using face detection, eye detection, and the proposed method. Figure(e) shows the detected lip area.

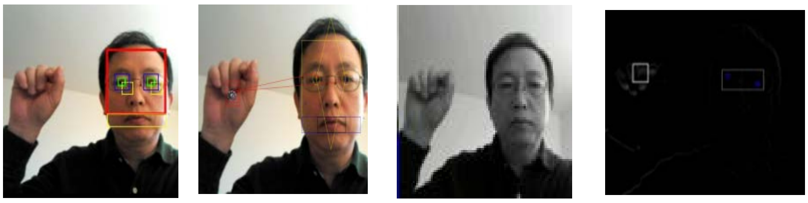
With the second proposed method (refer to the left eye, blue circle in Figure (d), (e)), we can detect the eye area along with the movement area. So it is possible to identify the movements of hands and arms and their directions and how much they move around.



(a) (b) (c) (d) (e)

Figure 7. Detection of Eyes and Lips and Augmented Image of Virtual Object

6.4. Experiment 4



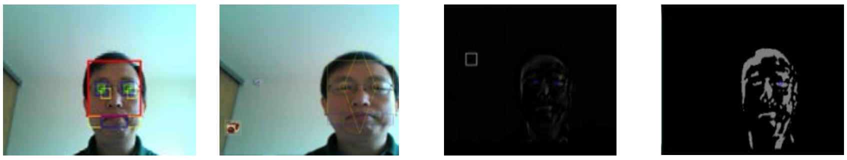
(a) (b) (c) (d)

Figure 8. Detection Image of Face, Eye and Lip and Resulting Image of Interaction with Virtual Object

In experiment 4, the gamer attacks the enemy character in AR environment. The purpose is to realize the interface for interaction with 3D objects in AR environment. When the hand collides with the enemy character, the eye ejects the energy laser to attack the enemy. Figure(a) shows the image of the hand attacking the spacecraft, the virtual object, after eye and lip detections were accomplished with the proposed method. In Figure(b), we augmented the enemy character in AR environment and the hand attacks it. The detection areas for the left and the right eyes are separated into green, yellow, and blue squares. The detection area for the lips is in yellow square. The enlarged face area is in blue square, and the detected lip area is in green. The rotating spacecraft, the enemy character, is in red square. Collisions appear according to the movements of the hand which is the interface. The detected right and left eyes are emitting energy beams and attacking the enemy character, the spacecraft.

Figure(c) is the grey image for the difference image. Figure(d) is the result of detection of the eyes and the hand movement area using difference image as proposed. With the traditional methods, the detection is only possible for the eye area where the eyes are on the same level(the Y-axis value). But with the proposed method, we can extract and process the movement data of the difference image. So the benefit is obvious; either for slanted shape or for the one on the same coordinate, the detection is possible. The white square on the left side of Figure(d) shows the area of the enemy character.

6.5. Experiment 5

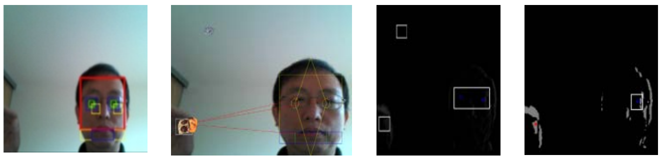


(a) (b) (c) (d)

Figure 9. Construction of AR Environment and Resulting Images of Eye, Lip, Objects

Different from Experiment 4, in Experiment 5, we augmented two virtual objects by adding a cubic object for interaction. This object is rotating on x and y axis and was produced with open GL ES. Figure(a) is the detection images of face, eye, and lip. Figure(b) is the image augmented with two 3D virtual objects. In addition, for difference image, in case there is noise in the movement information or if information volume is not enough, detection is less likely to be successful. So we proposed that we use mean filter in Figure(c) to augment the movement information of difference image as in Figure(d). The midpoint of the user's left eye movement in Figure(b) and Figure(d) is expressed in blue circle by using the filter output image of Figure(d).

6.6. Experiment 6



(a) (b) (c) (d)

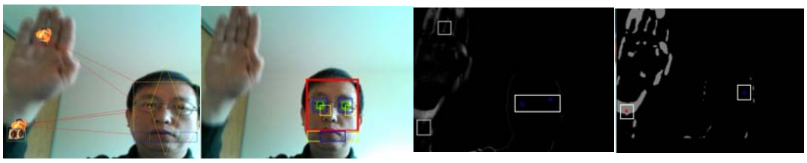
Figure 10. Detection of Eye, Lip, Hand on AR Environment and Resulting Image of Interaction Control

In Experiment 6, we tested the interface control for interaction with gamer. For that purpose, two 3D enemy character spacecrafts which were made in experiment 5 and a cube appear. Gamer attacks the cube with hand when the enemy character (Figure (b)) appears on the upper right and lower right to the gamer(user). At this point, we can infer the hand movements and the position information of the cube to determine whether there was a collision. We can see that in Figure(d) smoothing effect through mean filtering reinforces the movement information for better processing compared with the difference image in Figure(c). If you check out the image information of Figure(d), you can find that hand movement information(the red circle) and the analysis of the eye movement detected eye area exactly.

Since the hand collided with the cube, gamer's eye and the mouth eject the energy laser to destroy the enemy character. In Figure(c), on the other hand, only the difference image was used to analyze the movement information and the positional characteristics were considered to detect the eye and the two enemy characters. The right eye in Figure(a) corresponds to the information used in Experiment 5. The left eye, on the other hand, is different from the previous experiment because we considered the midpoint of the variation in Figure(d). The difference image has been processed to 'int type' for data compatibility.

When we applied the proposed algorithm to the two objects through Experiment 6, we could sense the collision on interaction as well as detect eyes and lips through face detection. Attacks with laser and explosion were naturally controlled and realized just like in experiment 6.

6.7. Experiment 7



(a) (b) (c) (d)

Figure 11. AR Environment and the Controlled Interaction with Two Objects

In Experiment 7, the user attacks the two enemy characters simultaneously. The spacecraft, the first character, is attacked by the hand.(Figure (a),(b)) The second character cube is attacked by the wrist. Attacks cause collision. The experiment was designed to show that the attacks with hand, wrist and mouth destroy the objects. With the application of the proposed algorithm, the user in Experiment 7 attacked the two objects at the same time; face detection led to detection of eyes and lips(Figure (c),(d)); the collision on interaction was sensed; attacks with energy laser and explosions were naturally controlled as in Experiment 6.

7. Conclusion

This paper proposes natural interaction control methods for constructing augmented reality on an android smart phone, using markerless tracking of facial and arm features. To do this, we have developed movement detection methods for eyes, lips, and hands and tested their effectiveness through several experiments.

We have used a method for detecting the eyes and lips by making inferences based upon facial detection information. Our method does not use traditional methods for image information extraction that depend on complex color model transformation. Rather we used the image difference between frames and augmented images.

Our method facilitates robust detection and tracking that is computationally efficient and more viable for implementing natural interaction with virtual objects in a mobile AR application. Further studies are needed to improve detection methods in environments with a lot of visual noise.

References

[1] J. Jeon, "Standardization Trend in Mobile Augmented Reality", TTA Journal, 01 - 02 /, vol. 139, **(2012)**, pp. 81-86.

[2] J. H. Jeon and S. Y. Lee, "Standardization for Mobile Augmented Reality Technology", Analysis of the Trends in Electronic Communications Book, vol. 2, no. 26, **(2011)** April, pp. 61-74.

[3] J. Chun, "Vision-based Motion Control for the Immersive Interaction with a Mobile Augmented Reality

Object", Korean Society of Internet Information, vol. 12, no. 3, **(2011)**, pp. 119-129.
[4] R.-L. Hsu, M. Abdel-Mottaleb and A. K. Jain, "Face Detection in Color Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, **(2002)** May, pp. 696-706.
[5] R. C. Gonzales and R. E. Woods, "Digital Image Processing", 2nd Edition, Prentice Hall, **(2002)**.
[6] homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm.
[7] http://blog.naver.com/PostView.nhn?blogId=budlbaram&logNo=50106543009.
[8] http://developer.android.com/reference/android/media/FaceDetector.html.
[9] http://www.developer.com/ws/android/programming/face-detection-with-android-apis.html.
[10] M. Kim and H. Han, "Face Identification Using Topological Relationship between Lips′ Axes and Eyes", Proceedings of the IEEK Conference, vol. 7, no. 4, **(2003)**, pp. 2028-2031.
[11] J. Jo Nam, "Adaptive Method of Eye Location for Face Recognition", Inha Univ., PhD. Thesis, **(2005)**.
[12] Z. Zhu, K. Fujimura and Q. Ji, "Real-Time Eye Detection and Tracking Under Various Light Conditions", ETRA'02 New Orieans Louisiane USA Copylight ACM 2002, **(2002)**, pp. 139-144.
[13] L. Weijun, X. Jiin and W. Shoujue, "A Fast Eye Location Algorithm Based on Geometric Complexity", Proceeding of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P. R. China, **(2004)** June, pp. 15-19.
[14] R.-L. Hsu, M. Abdel-Mottaleb and A. K.Jain, "Face Detection in Color Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, **(2002)** May, pp. 696-706.
[15] Y. Jae Lee, "Effective Interface Design Using Face Detection for Augmented Reality Interaction of Smart Phone", International Journal of Smart Home, vol. 6, no. 2, **(2012)** April, pp. 25-31.
[16] Y. Jae Lee and G. Ho Lee, "Augmented Reality Game Interface Using Effective Face Detection Algorithm", International Journal of Smart Home, vol. 5, no. 4, **(2011)** October, pp. 77-87.
[17] Y. Park, G. Kim and H. Choi, "Extraction Method of Skin Region using Skin Color of Eye Zone in YCbCr Color Space", The Korean Institute of Information Sciencetists and Engineers, vol. 17, no. 7, **(2009)**, pp. 520-523.

Authors

Thomas Furness received the B.S. degree in Electrical Engineering from Duke University in 1966, and the Ph.D. in Engineering and Applied Science from the University of Southampton, England in 1983. He is currently a Professor in the College of Engineering at the University of Washington and the founder of the international family of Human Interface Technology Laboratories at the University of Washington, University of Canterbury (Christchurch, New Zealand) and University of Tasmania (Australia). His research interests include virtual reality, augmented reality and human photonics.

Young Jae Lee received the B.S. degree in Electronic Engineering from Chungnam National University in 1984, the M.S. degree in Electronic Engineering from Yonsei University in 1994, and the Ph.D. in Electronic Engineering from Kyung Hee University in 2000. He is presently a Professor in the College of Culture and Creative Industry at Jeonju University. His research interests include smart media, computer vision, AR(Augmented reality), and computer games.