

New Interface Design Using Accelerator Sensor for Smart Phone

Young Jae Lee¹

¹*Dept. of Multimedia Jeonju University, 303 Cheonjam-ro, Wansan-gu,
Jeonju-si, 560-759, Korea
leeyj@jj.ac.kr*

Abstract

Touch screen interface is inconvenient since it requires both hands. So this study suggests new interface as replacement. Smart devices use built-in acceleration sensor. Low pass filter and Kalman filter reduce noise and variability of the output of the acceleration sensor to enhance precision and certainty, which is then applied in widgets or menus to test usability and functionality through experiments. Besides, sensor-based color detector is used to realize real-time color detection of the camera output image, and to search color information of the image, which is then analyzed and inferred for face detection. With the suggested algorithm, we can detect the face image even when it is slanted. Also, real-time detection of color information depending on the environment is possible. Experiments have proven its features which are particularly strong in the changes in the environment. The suggested algorithm is useful for further studies concerning various interfaces on smartphone.

Keywords: *interface, acceleration sensor, low pass filter, smart phone*

1. Introduction

Smart gadgets are getting popularized, with many kinds of smart phones or tablet PCs in wider use and the technologies on mobile sensors are advancing. Thus, there are a lot of studies going on about recognizing user behaviors based on smart devices. Especially in the case of smart phones, with various built-in sensors, the focus of many researchers is on developing convenient user interfaces [1-14].

For functional reasons, various sensors are added to the smart phone, which include acceleration sensor, magnetic field sensor, gyro sensor, proximity sensor, light sensor, temperature sensor, air pressure sensor, and etc. Acceleration sensor requires less energy and is also less subject to environment factors such as light or noise compared with other sensors, so it is possible to detect user's movements consistently. Technologies recognizing user movements through acceleration sensor are applicable in various fields. Better yet, along with gyro sensor, the acceleration sensor can display the information on movements and rotation of the smart phone in number terms, which facilitates many research efforts [1-9].

Motion gesture recognition through smart phone sensors is very useful in particular situations or environments. For example, touch screen interface on smart phone is convenient and user-friendly, contributing to widespread use of smart phone. But it is getting cumbersome since it requires both hands as smart phone functions are getting complicated and the screen size is getting bigger. As a solution, this study suggests a new interface using acceleration sensor on smart phone. The suggested method involves filtering of the sensor output information to get the exact and precise information, which is then used as movement information. Derived movement information can make the selection process easier on widget or web menus. The suggested method has an advantage of using only one hand over the conventional touch screen interface which requires both hands. In addition, the new interface can help detect the color information of the camera

input image real time. Especially when the subject is human, detection of face is possible using color information of the image in the face area. This trait is significant since detection of face is possible even when the face is slanted.

2. Sensors on Smart Phone

A sensor is a device which recognizes the information on a subject and transforms the information into a signal to send it to a physically separate location. The way human sensors work may be compared to information input in machine. The signals detected during input can be transformed into electric signals which then can be changed into computer-recognizable signals through amplification, accumulation, or tampering so that they can be suitable to each application purpose. There are many kinds of sensors which are principally used for mobile gadgets such as smart phone: micro(sound) sensor, light sensor, touch sensor, pressure sensor, temperature sensor, infrared sensor, acceleration sensor, gyro sensor, motion sensor, magnetic sensor, touch sensor, bio sensor and etc. [10-14]

Devices using Android OS have sensors which can measure various environmental factors like motions or directions. These sensors are capable of providing highly precise raw data, monitoring three dimensional movements or location of the device, or showing relative environmental changes nearby. [6-7]

3. Android Sensor

Devices using Android OS have sensors which can measure motions, directions, and various environmental factors. These sensors are capable of providing highly precise raw data, monitoring three dimensional movements or locations, or showing relative changes in the environment nearby. For example, we can infer complex movements or motions of the user-tilt, shake, rotation, or swing-from the gravity sensor. Weather application uses temperature and humidity sensors on the device to calculate and output dew point. Travel application uses magnetic sensor and acceleration sensor to output three dimensional directions just like compass, and gives information on the surroundings by working with the image information on the camera. Functional categorization of Android sensors can be divided into three parts as below. [4-11]

3.1. Categorization of Android-based Sensors

3.1.1 Motion Sensor: Motion sensor conducts three-axis measurement of acceleration force and rotation force. Motion sensor includes acceleration sensor, gravity sensor, linear acceleration sensor, gyroscope, and rotation vector sensor.

- Sensor.TYPE_ACCELERATER : acceleration sensor
- Sensor.TYPE_GRAVITY : gravity sensor
- Sensor.TYPE_LINEAR_ACCELERATION : linear acceleration sensor
- Sensor.TYPE_GYROSCOPE : gyroscope sensor
- Sensor.TYPE_ROTATION_VECTOR : rotation vector sensor

3.1.2 Environment Sensor :Environment sensor measures various environmental factors.

- Sensor.TYPE_AMBIENT_TEMPERATURE : ambient temperature sensor
- Sensor.TYPE_TEMPERATURE : temperature sensor
- Sensor.TYPE_LIGHT : light sensor
- Sensor.TYPE_PRESSURE : pressure sensor
- Sensor.TYPE_REALITY_HUMIDITY : relative humidity sensor

3.1.3 Position Sensor:Position sensor measures physical locations of the device.

- Sensor.TYPE_MAGNETIC_FIELD : magnetic field sensor

- Sensor.TYPE_ORIENTATION : orientation sensor
- Sensor.TYPE_PROXIMITY : proximity sensor

3.2 Sensor Coordinate

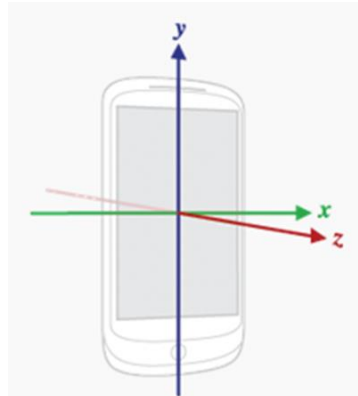


Figure 1. Sensor Coordinate System

Figure 1 shows the coordinate applied in magnetic field sensor, acceleration sensor, and gyro sensor. X axis is the horizontal part. Y axis is the vertical part, and the z axis is the front and the back part. The magnetic field sensor measures the size of the magnetic field corresponding to each axis. The acceleration sensor measures the degree of acceleration for each axis, and the gyro sensor measures the rotation angle for each axis[6-7].

3.3 Acceleration Sensor

Accelerometer measures acceleration in the direction of the movement. Three axis accelerometer measures static acceleration and dynamic acceleration toward x, y, and z axis. The static acceleration is measured as gravity acceleration imposed on the device in a static state, so we can get the incline. On the other hand, the dynamic acceleration is measured as speed variations while the device is moving, so we can measure movements. Since basically the acceleration sensor also measures the gravity acceleration, the output value is $-g$ (gravity acceleration) toward the z axis while the device is maintained in a horizontal state. Quick measurement is possible with acceleration sensor, but the response is not speedy enough when rotation angle is being measured. However, vector-type recognition is possible when movement occurs from one specific point to another specific point.

3.4 Magnetic Field Sensor

The magnetic sensor is appropriate when we want absolute information on directions. The navigation map uses the magnetic field sensor in the imbedded map to give exact directions. The measurement unit is micro-Tesla(uT), which is very precise, and its response is not very good since it is heavily influenced by the environment.

3.5 Gyro Sensor

Gyro sensor is used to measure the rotational angular velocity in X, Y, and Z axis. The rotation axis for controlling gyro sensor is called 'Pitch' in the x axis, 'Roll' in the y axis, and 'Yaw' in the z axis. Low credibility of measurement is the major drawback, but the measurement is quick, and it is also possible to measure rotational angular speed.

4. Filter

4.1 Low Pass Filter

LPF(Low Pass Filter) allows signals with low frequencies and filters out signals with high frequencies. Generally, LPF has the primary purpose of eliminating noise. Formula(1) shows LPF with alpha value(weighted) added. In LPF, we need to notice that in the average filter formula, weighting is differently applied for average and measurement value.

$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1 - \alpha)x_k, 0 < \alpha < 1 \quad (1)$$

\bar{x}_k is the filtered value, \bar{x}_{k-1} is the previous average value, and x_k is the newly added data.

In the formula above, the weight ‘ α ’ is the value adjustable to the environment. If α is close to 1(0.7 or 0.8, for example), the noise is big. If the α value is small, the noise is increased, but the delay time is reduced. That is to say, if we increase the ‘ α ’ value, it means we give a lot of weight to the prior value, thus making the signal smooth. When ‘ α ’ is small, the filtered result is close to the original signal [6-7].

4.2 Kalman Filter

Kalman filter is the optimal inference method which uses the probability model of the subject system along with the measurement value to find the state variable of the system[8-9][14]. In other words, the state value is inferred over time and the estimate is updated using a state transition model and measurements. This algorithm minimizes the error covariance between the actual state value and inferred state value. Kalman filter is operated based on the linear system and the white noise with standard distribution(input noise(driving noise) and sensor noise (observation noise) is white noise). It is known to be the optimal unbiased inference system with minimal covariance. Kalman filter considers the probability distribution of the inferred value and selects the value with the highest probability. It calculates the final inference value by adjusting the predicted value based on the prediction errors of the measured value. The objective of the Kalman filter is to find the optimal average factor for each resulting state of the Kalman gain.

Initial estimates for \hat{x}_{k-1} and P_{k-1}

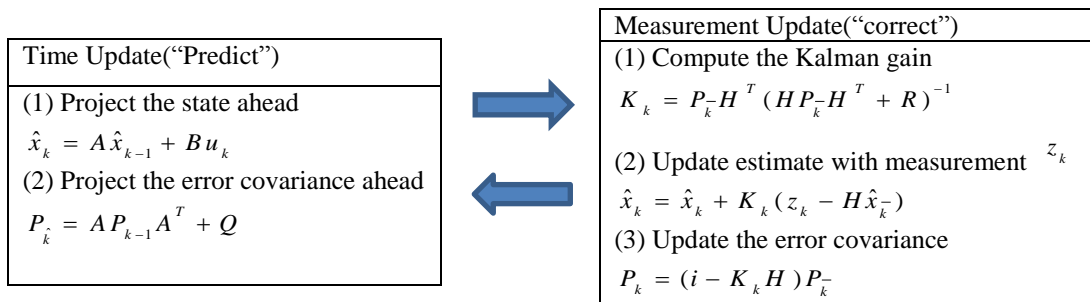


Figure 2. Overall Flowchart of Kalman Filter

Figure Q shows the overall flowchart of Kalman filter in which time update moves to measurement update. \hat{x}_k , the value gotten from measurement update, becomes \hat{x}_{k-1} in the next time updater. When selecting the first value, we input the predicted error covariance which the first state variable \hat{x}_{k-1} and state variable P_{k-1} will have. In the Time update, variable P and predicted error covariance P_k^- are predicted using the formerly calculated P_{k-1} , system model A, and system error Q. The resulting error

covariance P_k is used to calculate Kalmangain(K). To get the estimated value, the difference between z_k (currently measured sensor value) and \hat{x}_k^- (the value gotten from time update) is multiplied by Kalman gain, and then \hat{x}_k^- is added to decide \hat{x}_k . Error covariance is the measure for the accuracy of the estimated value \hat{x}_k . By referring to this error covariance, we can decide whether to use the previously calculated estimation value or not. Kalman filter is based on the system model, and helps predict the next state error and error covariance to compensate for the difference between the measured value and the predicted value.

5. Suggested Algorithm

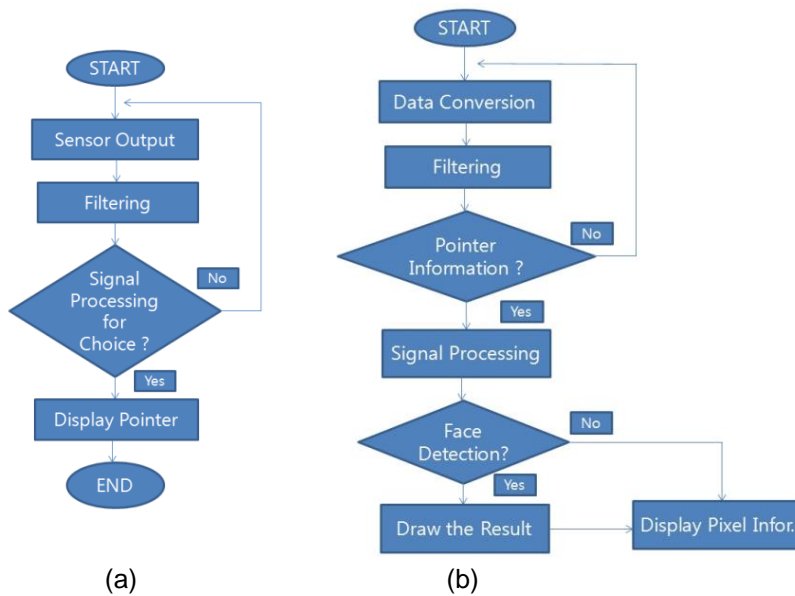


Figure 3. Flow Chart of the Proposed Algorithm

Figure(a) is the configuration of the suggested algorithm when selections are made with the pointer on app or menu. Sensor output is input, and filtering is operated to reduce variability and noise in the sensor. Depending on each purpose and usage, we can select either Kalman or Lowpass filter. The filtered signal is checked for value variation on each axis. Location data(as in temporary motionless state) and movement data are analyzed to extract the displacement information through inference. The extracted information is applied to the pointer, which is transformed to select menu or app on the screen. The suggested method can replace the conventional touch screen. Through experiments, we test functionality and usage.

In Figure(b), the suggested pointer is used to extract the image color information from the camera output in real time. The extracted information goes through image processing to detect face. The input camera image(YUV420SP) is transformed into RGB image and filtering is conducted to reduce noise in the original image. At this time, we need to decide about color extraction after checking the information on the location and movement of the pointer. If color extraction is necessary, we check the color information of the image of the current location of the pointer. This kind of information can be used for detection of other objects as well as a specific subject like face or hand. We compare the detected color information with the input image information. If the two are of the same color, we mark it. Face detection is possible when the area is detected which has the similar color with the face. Remarkably, even the slanted face can be detected. The suggested algorithm(b) is also tested for its performance and validity through experiments.

6. Experiment

6.1 Experiment 1

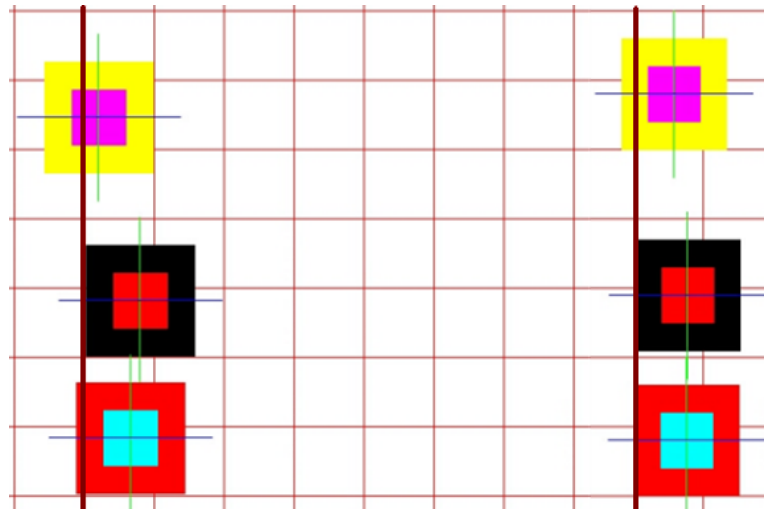


Figure 4. Objects Position of Experiment 1

In Experiment 1, the output of the acceleration sensor is shown in three ways. The first yellow square at top(R1, the inner square is pink) shows the output of acceleration sensor without filtering. The second black square(R2, the inner square is red) shows the output where Kalman filter has been applied to the sensor output. The third red square(R3, the inner square is sky blue) is the resulting output where Lowpass filter was applied to the sensor input. In Figure(b) on the right hand, the squares are initialized and rearranged with the red line on the left as center. Here, the sensor output square R1 is trembling since it is not appropriately arranged due to continuous changes in value. We can see that R2(Kalman filter applied) and R3(Lowpass filter applied) are well arranged without tremors with the starting line as center. Figure(a) shows the result image where the movement proceeded to the goal line in space 8. In Figure(a), in R1, we can see movements with many errors around big lines arranged on the left. In R2 where Kalman filter was applied, arrangements are neat, safely reaching the goal line. In R3, where Lowpass filter was used, there is some movement displacement to the front, but still the arrangements are neat as in R2. Through experiment 1, we could verify that filtering of signal is necessary due to variability of sensor output. Kalman filter has an advantage of safely and precisely reaching the goal line, but it is slow since it involves large volume of calculations. When Lowpass filter was used, there was some displacement, but the response was quick. Therefore, when we need precise and accurate control, Kalman filter is appropriate. On the other hand, when we need quick and real time response, Lowpass filter is more apt.

6.2 Experiment 2

Based on the results from Experiment 1 , Experiment 2 used acceleration sensor and highly mobile and compliant lowpass filter instead of the touch screen method for the purpose of testing new menu setting methods. The menu used for the experiment is the one for customer registration, which is one of the most widely used applications for customer dealings. For the purpose of the experiment, three kinds of interactions were used as the menu points which include gender, ways to receive user information and process to send the information to the server, with 'name' omitted from the suggested user

information.

6.2.1 Experiment 2.1



Figure 5. Customer Registration of Experiment 2.1

In Experiment 2.1, rocket is used as the menu pointer for movement. (Figure(a)) When there is nothing selected, 'send button' acts as the menu pointer and when it is clicked, we get the output comment "No choice." (Figure(b)) In Figure(c), the menu pointer approaches the twitter check box in 'Reception'. When we touch the middle part of the twitter button properly, it is sensed that the twitter is touched, so the signal notifies the user that the selection was made to yellow from red. (Figure(d)) To send the information to the server, we click the send button on the menu pointer, and the result is output into text. (Figure(e)) Through Experiment 2, we could see that the suggested methods worked well for the menu pointer. Especially, mobility and compliance, the key features of the lowpass filter were well utilized to prove that the suggested new methods using acceleration sensor, filter, and menu pointers worked successfully for the menu settings replacing the conventional touchscreen method.

6.2.2 Experiment 2.2

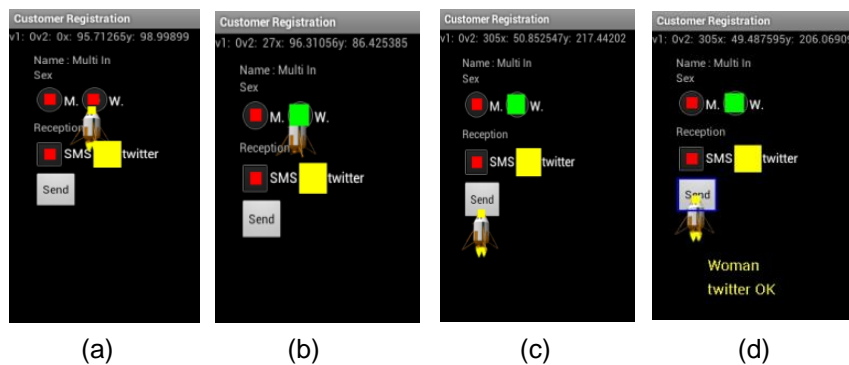


Figure 6. Customer Registration of Experiment 2.2

In Experiment 2.2, gender was added to the results of Experiment 1. In Figure(a), there is movement to select 'gender' after the twitter is set. In Figure(b), Woman(W) is selected. Figure(c) shows the movement to 'send button', which is to send the information to the server. Figure(d) is the resulting text output of 'gender' and 'reception' after sending.

6.3 Experiment 3

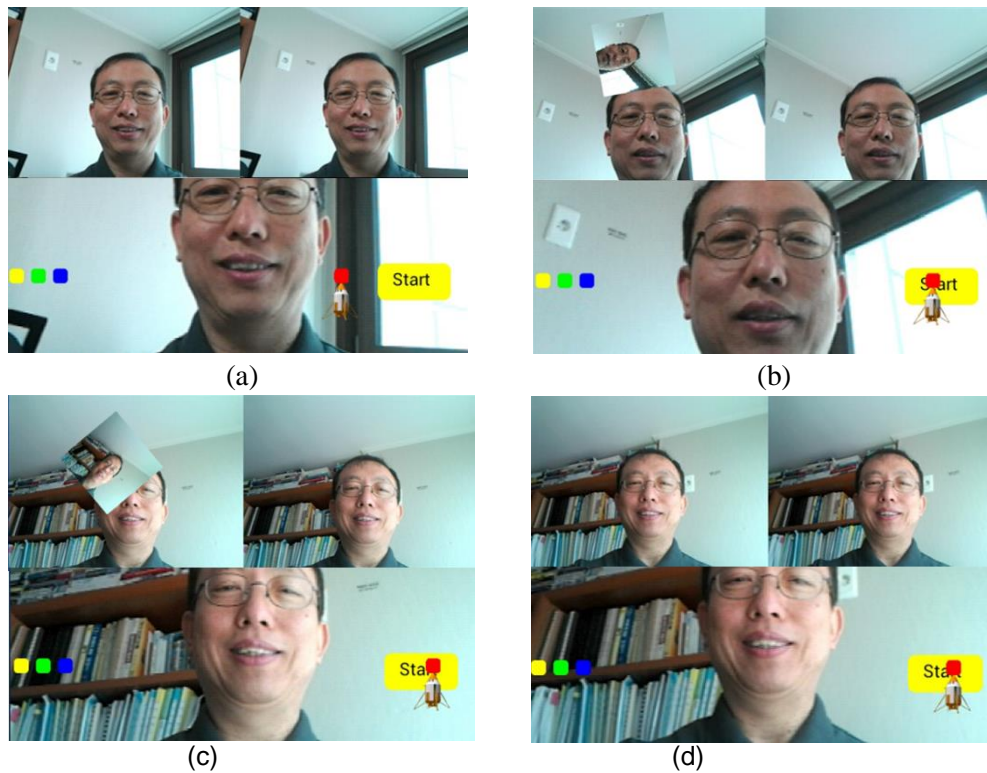


Figure 7. Start Button Test Using Rocket Pointer

Rocket pointer is used to test the on/off functions of the start button. To distinguish on and off states, we check whether, at a particular location of the pointer, there is no sensor output for a certain time on a certain point on x and y axis. With the resulting image, the current camera input is created into a new object to realize moving and rotating animation on screen. Figure(a) shows the pointer on the move. In Figure(b), the on state, the pointer stays at the location of the Start Button for a certain time. Here, the image currently input on the camera is transformed into rotating and moving animation output. The resulting image is realized here. In Figure(c), the start button is also on as in Figure(b), and the animation image is realized. In Figure(d), the start button is off and the resulting image does not show any rotating or moving animation. We could confirm that the start button gets easily activated with the use of the pointer.

6.4 Experiment 4



Figure 8.Face Detection Button and Rocket Pointer

In experiment4, to detect the face, we press ‘the face detection button’ with the pointer. Move the pointer to press the blue button on the left bottom to make it ‘on’(refer to the picture below Figure(a))(the pointer presses on the blue button, the color changes into the red color of the pointer), and the face is detected(refer to the picture at the upper left of Figure(a)). The button used in the experiment 4 is smaller than the ‘start button’ used in the previous experiment, so more precise and accurate movement is required. With the face recognition method in Figure(b), we could identify the bitmap size of the input image through ‘FaceDetector’ and ‘FaceDetector.Face class’ provided on Android[15], and location and number of the face existing within the image are also identifiable. In addition, provided are CONFIDENCE_THRESHOLD constant(set at 0.4) and EULER value of each face on x, y, and z coordinate(set at x=0, y=1, and z=2). With the provided information on values, we can get the mean point of the distance between the two eyes, with which the eye location can be calculated. This method requires relatively low amount of calculation and thus is relatively stable. Therefore, it is applicable when quick and stable condition is required like in AR(augmented reality) environment[11-15]. By using the pointer proposed in experiment 3, application is possible even where accuracy and preciseness is required.

6.5 Experiment 5

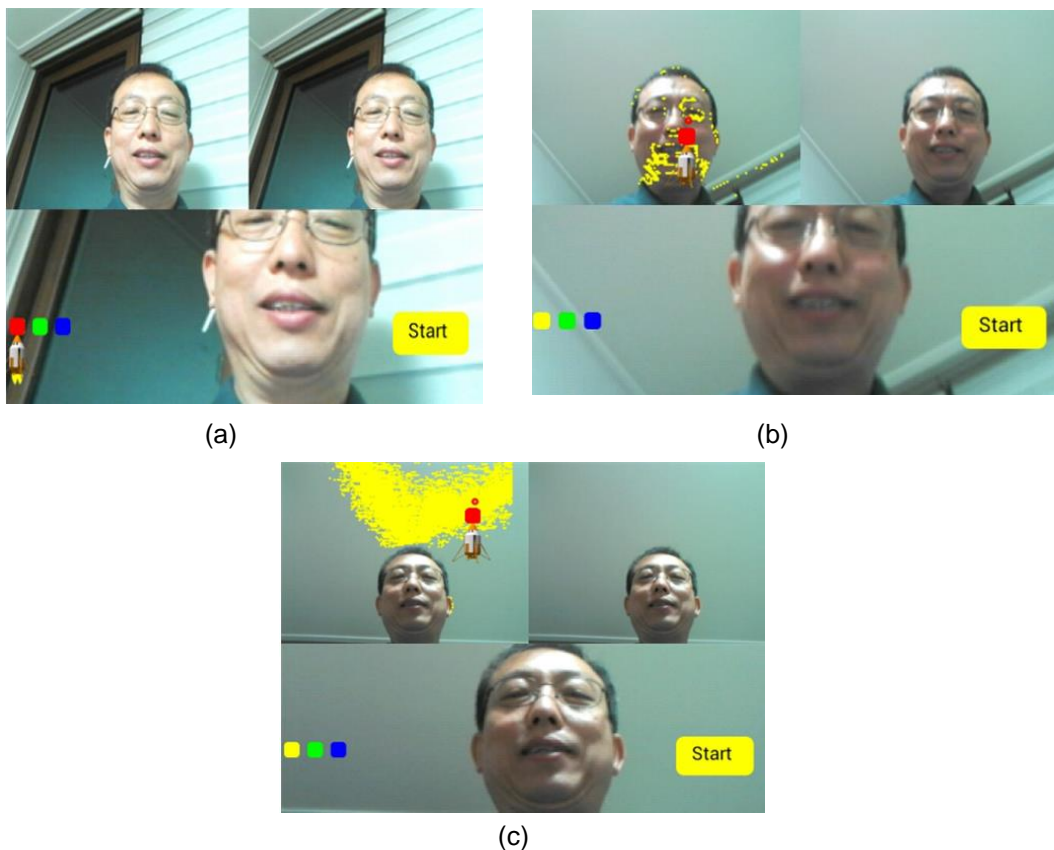


Figure 9.Color Information for Region of Interest Using Rocket Pointer

In the experiment 5, with the suggested method, we extract the color information for the required area in the camera input. Figure(a) is the input image. In Figure(b), we first read the color value around the eye area(red small circle), and then mark the same color information within the input area in yellow circle. We can see that most of the area around

the face is showing similar color. Figure(c) shows the image of color extraction of particular area at the ceiling. The red circle at the end of the pointer extracts the similar color information, which is marked in yellow circle. In the Experiment 5, through the resulting image, we could realize the color information of a certain area in the camera output image by using the pointer. The resulting image from the suggested algorithm was also well expressed and successfully output.

6.6 Experiment 6

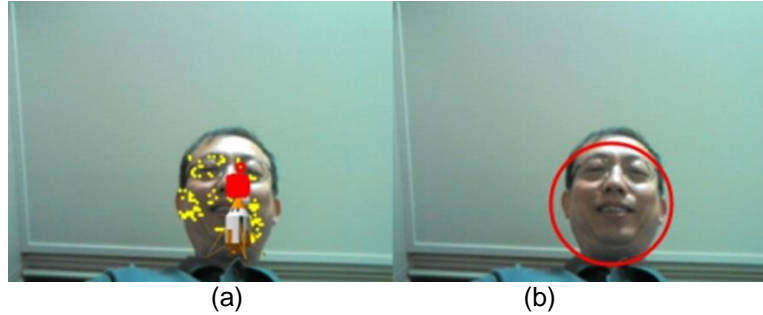


Figure 10.Face Detection Using Color Information of Rocket Pointer

In experiment 6, face recognition was tested with the proposed method. In Figure(a), the proposed pointer is moved to be located precisely where the face is. Color information of the face is also extracted using the pointer. With this information, the proposed method detects the face area. Figure(b) shows the extracted result represented in red circle. Experiment 6 showed the successful extraction of face area with the proposed algorithm.

6.7 Experiment 7

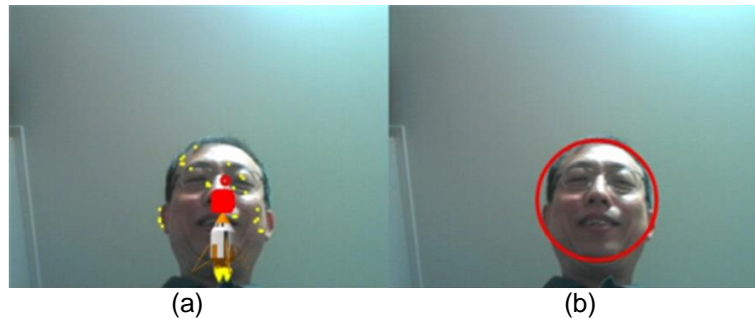


Figure 11.Face Detection Using Rocket Pointer Under Different Environment

The camera input image is changed due to the light source on the left. Images are easily influenced by light, so the experiment was designed in similar conditions. In Figure(a), there is influence of the light source and the proposed algorithm is used to extract the image information by using the pointer. Figure(b) shows the extraction of the face area with the prior information. Just like experiment 6, experiment 7 showed that the proposed algorithm is successful in extracting the face area.

6.8 Experiment 8

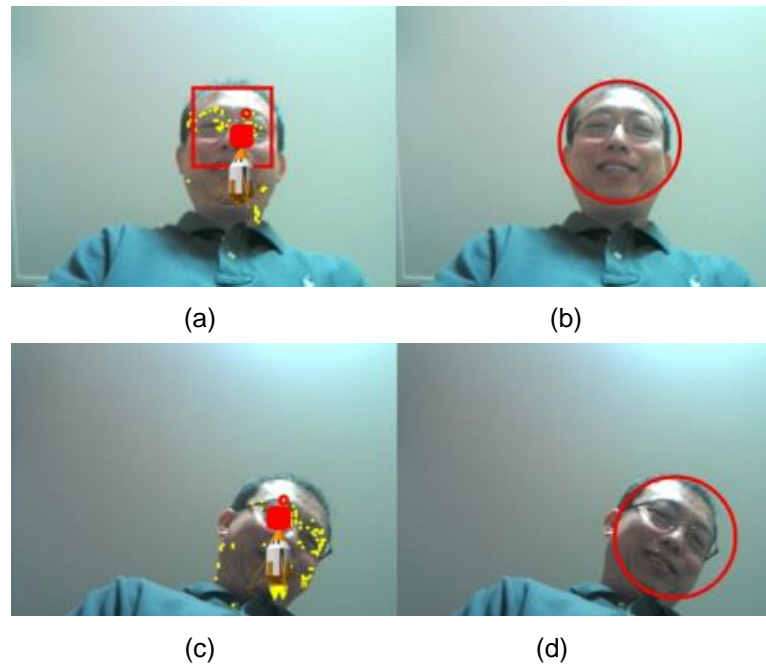


Figure 12.Face Detection of Experiment 8

In experiment 8, Figure(a) shows the extraction of face using Android API. Figure(b) shows the extraction of the face area using the proposed algorithm after the color information was detected in the face area using the pointer. As we can see in Figure(a) and (b), detection of face area is successful in both cases. However, when the face is slanted as in camera output image Figure(c), face detection was not possible using Android API. But in Figure(d), when the suggested algorithm was applied, the face detection was successful. As experiment 8 reveals, with the proposed algorithm, face detection is successful even when the input image is slanted.

7. Conclusion

When using mobile devices with the touch screen technology, we usually need to hold the device with one hand and touch the screen with the fingers in the other hand. However, as mobile devices are getting more functions, and their screen gets larger, it is impossible to hold the gadget in one hand. Furthermore, new products are emerging which engage both hands. Touch screen interface is simple and convenient, and it is popular among all kinds consumers regardless of gender or age. Actually it is one of the most widely used interfaces on the latest mobile devices including smart phone. However, the technology is cumbersome since it requires the use of both hands. To solve this problem, this paper proposes a new interface using acceleration sensor. Its performance was tested through experiments. In particular, filtering methods were suggested which can minimize mobility and noise of the sensor output as well as maximize accuracy and precision. Functionality and usability of the filtering methods were tested and confirmed through experiments by applying the methods in widget menu setting. The proposed interface was applied to color extraction pointer which showed the possibility of precise control. It was also possible to extract the face by using the proposed image processing method in pointer output. In particular, it was possible to extract the face even when the face was slanted. Also even in case where color information is changed due to the influence of light, the proposed color information pointer could easily extract the color information real time. So we could prove that the method is enduring and adaptable to changes in the environment.

References

- [1] J. H. Kim and I. C. Kim, "Design and Implementation of a Two-Phase Activity Recognition System Using Smartphone's Accelerometers", KIPS Tr. Software and Data Eng, vol. 3, no. 2, (2014), pp.87-92.
- [2] M. S. Park, S. H. Kang and O. Chae, "Robust Hand Detection and Tracking using Sensor Fusion", The Korean Institute of Information Scientist and Engineers, Journal of Software and Application, vol. 40, no. 9, (2013), pp. 558-556.
- [3] S. H. Nam, J. H. Kim, S. K. Heo and I. C. Kim, "Smartphone Accelerometer-Based Gesture Recognition and its Robotic Application", Korea Information Processing Society, Journal of Software and Data Eng., vol. 2, no.6, (2013), pp. 395-402.
- [4] Y. C. Lee and C. W. Lee, "Android Platform based Gesture Recognition using Smart Phone Sensor Data", Journal of Smart Media, vol. 4, no. 14, (2012), pp. 18-26.
- [5] [<http://terms.naver.com/entry.nhn?docId=923318&cid=3430&categoryId=3430>]
- [6] Jun Won Lee, Android Sensor Story, Freelec Press, (2014).
- [7] http://developer.android.com/guide/topics/sensors/sensors_overview.html
- [8] <http://blog.naver.com/hms4913?Redirect=Log&logNo=30166486895>
- [9] <http://gyjmobile.tistory.com/>
- [10] H. J. Lee and D. H. Chung, "Difference in Interactivity, Flow, Mood, Attitude, and Intention According to Smartphone Game Sensor", Korean Association for Broadcasting & Telecommunication Studies, vol. 26, no. 1, (2012), pp. 126-166.
- [11] Korea Electronics and Telecommunications Research Institute, "Sensor technology/ market research: part technology", (2001).
- [12] C. J. Young and C. B. Dong, "Technology trend of silicon-based pressure sensor", (2005), pp. 1-13.
- [13] D. M. Kim and C. W. Lee, "Technology Trends of Smartphone User Interface", The Korean Institute of Information Scientist and Engineers, vol. 28, no. 5, (2010), pp. 15-26.
- [14] <http://blog.naver.com/msnayana/80106682874>
- [15] <http://developer.android.com/reference/android/media/FaceDetector.Face.html>

Author



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 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.