School of Professional Education and Executive Development
專業進修學院

SPEED

HONG KONG COMMUNITY COLLEGE
香港專上學院

Working Paper Series

Title	3D medical image processing through multiple mobile devices
Author(s)	Ken K.L. Peng
Issue Date	2015
Issue Number	3
Paper Number	5
Citation	Peng, K. (2015). *3D medical image processing through multiple mobile devices* (Working Paper Series No. 5, Issue 3, 2015). Hong Kong: The Hong Kong Polytechnic University, College of Professional and Continuing Education, School of Professional Education and Executive Development. Retrieved Jan 5, 2016 from http://weblib.cpce-polyu.edu.hk/apps/wps/assets/pdf/w20150305.pdf
Rights	Copyright for this paper is owned by the author(s).

CPCE Disclosure Authorized

School of Professional Education and Executive Development
專業進修學院
SPEED

HONG KONG COMMUNITY COLLEGE
香港專上學院

Working Paper Series No. 5, Issue 3, 2015

3D Medical Image Processing through Multiple Mobile Devices

Ken K.L. Peng
School of Professional Education & Executive Development
College of Professional & Continuing Education
The Hong Kong Polytechnic University
Kowloon, Hong Kong
kenpeng@speed-polyu.edu.hk

ABSTRACT

Considering the trend of using mobile devices for 3D medical image viewing, we constructed a cloud service system for 3D medical image processing using mobile devices. A multi-plane 3D display system was built that allows user to see and interact with 3D image via multiple mobile devices. 3D images are constructed by stacking continuous 2D image data other than the traditional 3D object. The cooperation of mobile devices brings out a special feature, cascaded view, which has a property that cutting faces of different mobile devices are concerned. Considering user experience, bookmark and annotation functions were provided for user to return to specific cutting face state and make note on it as picture archiving and communication system (PACS). We made a service prototype using computed tomography image as resource of 3D image for proof of service. Methodology of Service experience Engineering (SEE) was applied to test the satisfaction of users. Results show that both doctors and patients are satisfied with our prototype of 3D medical image processing systems. Conclusions are suggested that the proposed 3D medical image processing systems could be helpful for future healthcare.

KEYWORDS: Medical image processing, Mobile devices, Clinic, Hospital information system

1 INTRODUCTION

In 3D visualization field, virtual reality is already a widely used technology to provide user 3D illusions and user can interact with 3D object by wearing special designed equipments. For example, in immersion virtual reality, user can see 3D scenes by wearing a head-mounted display or standing in a room with surrounding digital projections, and interact with 3D object by wearing hand motion capture gloves or equipping with other devices that have sensors inside for motion detection. Besides, in recent years, some people even invented devices that combine 3D display and motion capture in one device, such as pCubee (Stavness, Lam, & Fels, 2010). However, the above technologies have two common disadvantages. One is that the hardware requirement is too difficult for usual people to reach. The hardware is either too expensive or too rare to get. Moreover, pCubee is not even a reachable device for usual people except the inventor. The other disadvantage is that either user has to wear heavy wired equipment or they need big rooms for display. The lack of mobility makes user cannot see 3D images anywhere anytime. On the other hand, because of the rapid development of mobile devices in hardware that nowadays mobile devices are equipped with various sensors, it is possible for mobile device to be the medium of 3D visualization and interaction, and because of fast growth in population of mobile device user, mobile devices have become pervasive devices for people's daily use. As a result, in this study, we consider mobile devices as our display and interaction devices.

Methods mentioned above employ traditional visualization technique to render 2D images of the 3D object. The traditional visualization consists of two main steps. First, construct the 3D model of the 3D object. The 3D model is created by a set of points in 3D space, which are connected by geometric data such as lines. Second, render the shape of 3D objects into 2D images according to user's view point. However, in some cases, cutting faces of objects are more valuable than shapes of objects. For example, in medical image condition, the cutting face of the body is more important than appearance of the body. Therefore, we propose a multi-plane 3D display system that employs a novel rendering method for constructing cutting faces of 3D object via multiple handheld devices, and we apply computed tomography image, which is a widely used medical image type as our 3D image source for better understanding. In our rendering method, we use magnetic sensor data and accelerometer data to obtain the orientation of mobile devices and use the data to compute cutting faces.

In mobile 3D field, mobile devices are used mostly as a controller with display screen to interact with the 3D games. One user having one mobile device is sufficient for any 3D game, however, there is limited studies revealed the possibility of new interesting interaction patterns can be achieved by utilizing multiple mobile devices. Therefore, we offer user an entirely different experience about interacting with 3D object. We developed multiple handheld devices that can interact with each other and cooperate to see cutting faces at different positions in 3D objects. For position determination, we provide two localization methods, one is touch-based localization method, which offers an intuition way that user can touch the location he/she wants for the cutting face on screen of handheld device, and the other is barcode-based localization method, which take advantage of quick response code (Rajeeva & Sagar, 2014) to encode location information in images and allows user to get location by using camera to scan the code. We also provide a relocation method for user to move location by some touch gestures.

2 LITERATURE REVIEW

It has become a trend that everyone has mobile devices such as smart phone and pad. Demands for 3D graphics on mobile devices then appear. For example, users can play 3D games on mobile device and change game scene by rotating the device. As a result, user experience about interactions with 3D image also has become a popular issue. More and more works discuss about 3D interactions on mobile devices. Nowadays, a variety of sensors including accelerometers, orientation sensors and gyroscope sensors has become basic equipment in mobile devices; therefore, many works take advantage of different sensors to generate various 3D interaction commands to manipulate 3D object or change user's point of view in the game scene. For instance, Chittaro and Ranon (2007) used accelerometer to manipulate 3D molecular model. They provide a system that they can rotate 3D molecular model by leaning a mobile device and change viewing location by moving a mobile device. The accelerometer in system will detect the leaning direction and rotate the molecular model in the same direction. Hürst and Helder (2011) provided two visualization concepts and use different sensors for each concept. One is shoebox visualization and the other is fixed world visualization. In shoebox visualization concept, graphics are changed based on accelerometer to create illusion of a box attached to the mobile device. On the other hand, in fixed world visualization concept, graphics are changed based on orientation sensor to create illusion of a box surrounding the user. Moreover, they also provide touch methods to navigate or select objects in the scene. Hansen, Eriksson, and Lykke-Olesen (2006) used orientation sensor in the mobile device acted like a controller when user want to navigate in virtual reality environment on a large display, they can tilt the phone for viewpoint transforming. The tilting is detected by orientation sensor. Besides, they also designed touch interactions to control avatar moving and viewing camera zooming. Moreover, Geen and Krakauer (2003) also used mobile devices as controllers to manipulate 3D objects shown on a large display. They used gyroscope sensor to detect moving and rotating for users to move and rotate objects. However, user's point of view can be speculated not only by utilizing sensors' information but also by face tracking. Thus, Francone and Nigay (2011) computed the position of the device according to the user's head and use it to control the viewpoint on a 3D scene, but without using sensors, only applying face tracking has a disadvantage that user cannot see 360 degree of the scene. Presently, although the amount of studies about 3D visualization is increasing fastly, but they are mainly focus on one mobile device. There is no work discusses about more than one mobile device condition. Following works explore various interaction ways between mobile devices. Some of the studies develop information sharing systems that user can interchange information among mobile devices and public displays through some interaction methods. Francone and Nigay (2011) allows users to interchange digital information among their portable computers, table and wall displays, and other physical objects through hyper-dragging. A proposed interaction technique of hyper-dragging is that users can easily share information like a picture or video by using cursor to drag them to the physical place where they want to upload the information(Rekimoto & Saitoh, 1999). As a result, tables and walls can be seen as a spatially continuous extension of personal portable computers. Their work has an embedded infrastructure including an embedded control surface and two embedded displays. They provide a collaborative workspace that users can connect their personal mobile devices with the workspace through network, and share information or their personal screen with others by manipulating the control surface. Butler, Izadi, and Hodges (2008) proposed a pairing method to solve the problem of identifying who is interacting with a multi-user interactive touch display in multiple mobile devices situation. They use a depth camera to track user' position and associate each mobile device with a particular user by analyzing accelerometer data on mobile device. Then, body tracking and touch contacts positions are compared to associate a touch contact with a specific user. After these identification processes, users can interchange information among personal mobile devices and the public touch screen. Moreover, Kray, Nesbitt, Dawson, and Rohs (2010) investigated whether gesturing with a mobile phone can help to perform complex tasks involving two devices, and recommended several possible techniques for gesture recognition. For example, measuring signal strengths or runtime differences between signals to estimate distance between mobile devices, and use it to recognize devices' approaching or pulling away gestures. On the other hand, Hinckley (2003) provided a different gesture recognition method to recognize bumping between mobile devices. Users can tile together the displays of multiple tablets just by physically bumping a tablet into another one lying flat on a desk. The tiling is detected by accelerometer. However, studies about interaction between mobile devices have not concerning 3D visualization and 3D interaction in their works. Thus, this study aims to fill the gaps mentioned above.

3 SERVICE EXPERIENCE EGGINEERING

The current research applied an experimental design to determine the service of 3D medical image processing through multiple mobile devices by using methods of service experience engineering (SEE) (Lin et al., 2013; Peng, Chang, Tseng, & Deserno, 2012). SEE is a potentially useful and easy to implement technique from user expreiences for developing new services that really satisfy user's needs. The method concludes with a number of practical steps to ensure continuous improvements in new services (Hsiao & Yang, 2010). Quality fucntion deployment (QFD) was applied to acquire the details of service requirements, functions, thresholds and advantages of our 3 D medical image processing through multiple mobile devices (Lin et al., 2013; Peng et al., 2012).

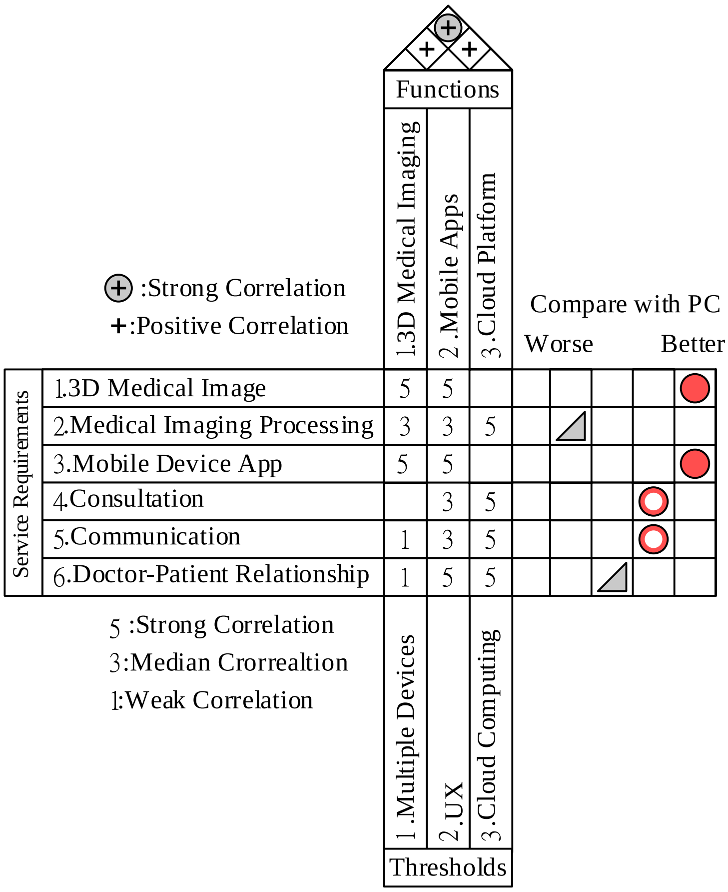

Figure 1. Quality Function Deployment of 3D Medical Image Processing

3.1 Basic Scenario

Comparing to existing medium, we present a novel method to interact with 3D object using handheld devices. The 3D object is placed in front of users as a virtual 3D image in the air, and users can use handheld devices such as tablet personal computers or smart phones to interact with them and see cutting faces. As shown in Figure 1, we use a 3D human head as an example, and we have a tablet and two smart phones in our environment. First, user can use a tablet putting on a desk to see a horizontal cutting face of the human head, and then use another handheld device, a smart phone (Fig. 2 ②), standing on the tablet to see a vertical cutting face relative to the cutting face on tablet by obtaining the location on tablet's screen. After that, user can also use another smart phone (Fig. 2③) to see other cutting faces with location obtained from the smart phone 1 and with different geographical orientations.

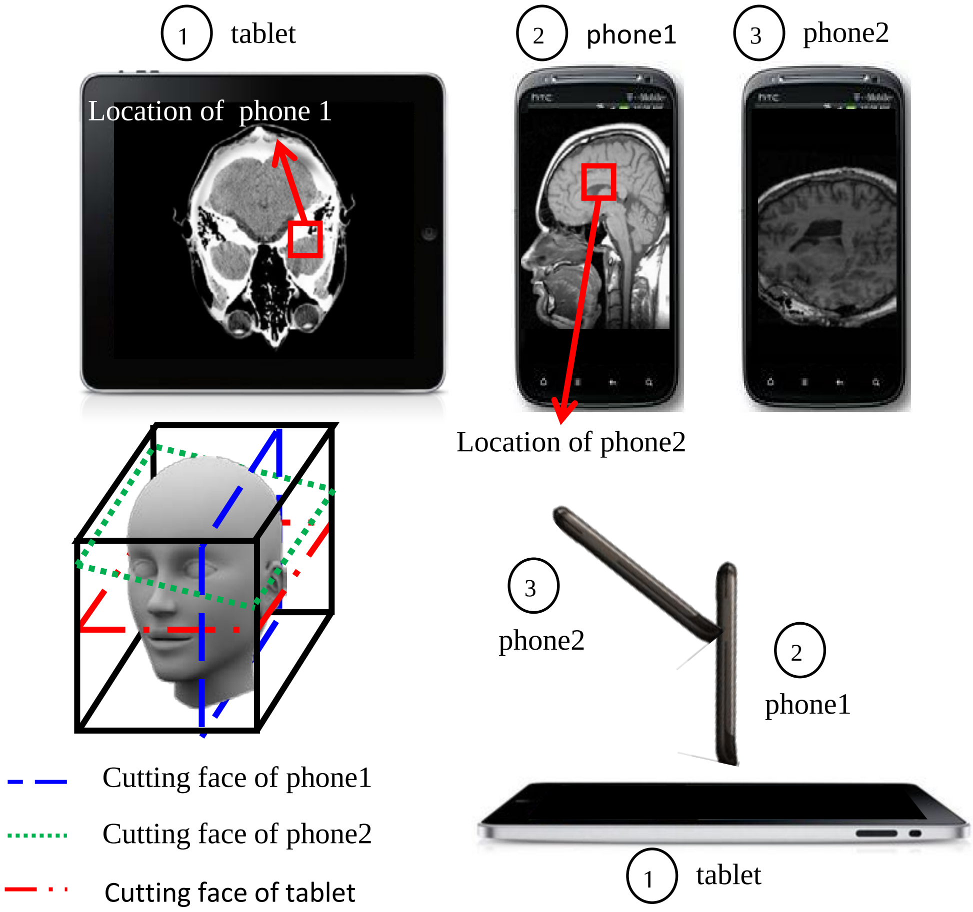

Figure 2: A basic scenario of interaction

3.2 System Architecture

To achieve the novel 3D visualization and interaction with 3D image via multiple handheld devices, we proposed a multi-plane 3D display system. Below is our system architecture.

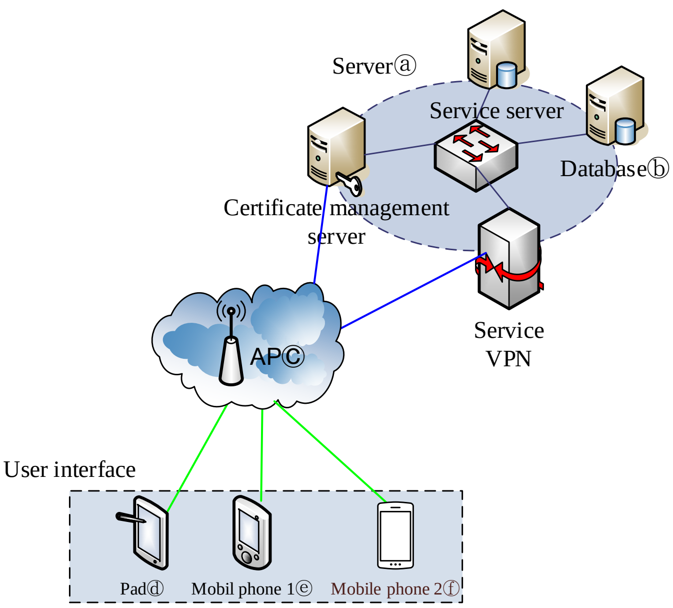

Figure 3: System architecture

Our system employs basic client-server architecture. The server side contains a server (fig 3) and a database (fig 3ⓑ). The server is responsible for all computing work include processing raw data of 3D object, constructing cutting faces, and cutting faces' location computation. The other component of server side, the database, stores the raw data of 3D image, bookmarks, and annotations. On the other hand, the client side is a tablet PC (fig 3ⓓ) or smartphones (fig 3ⓔ, fig3ⓕ), termed d_i, and each handheld device can connect to the internet through a wireless access point (AP; fig 3ⓒ) and communicate with our server. Mobile devices used in client side are responsible for motion capture to detect the orientation of cutting faces, cutting faces' location obtainment, and provides a user interface for displaying cutting faces and commands' input such as requesting a cutting face.

3.3 Software Architecture

According to the hardware architecture, the software architecture of our system consists of two parts. One part is server platform, and the other part is client platform. Each platform is divided into three layers, including hardware layer, middleware layer and application layer. For clarity of software architecture, the modules in different layers are in different colours. Modules in hardware layer are in blue, in middleware layer are in orange, and in application layer are in blue-green. The software architecture of server platform is shown in Figure 4. In our system, we provide three major functionality including rendering cutting faces, locating cutting faces, storing cutting faces as bookmarks and annotating on cutting faces, and according to the order, these functionalities are implemented by "Cutting face rendering" module , "Localization" module and "Bookmark & annotation" module in application layer. Regarding middleware layer, "Data loading" module will load in the raw data of 3D object from database and construct the 3D image for the use of rendering cutting faces. For example, in medical image case, 2D medical images are stored in database, and "Data loading" module will load in the images and construct the 3D medical image at start-up of the system. In hardware layer, "Wifi" module is in charge of client connections' establishment and maintenance, and network communication including request acceptance and corresponding response sending. More specifically, the "Wifi" module will classify the requests sent from client and forward them to appropriate application module for further handling and send back the result in response. For instance, client may send a request to ask for cutting face, then "Wifi" module will forward the request to "Cutting face rendering" module and send back the cutting face to client after rendering.

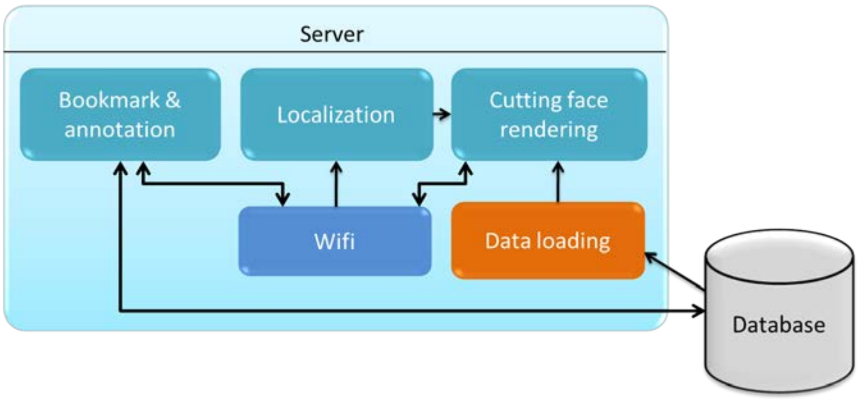

Figure 4: Software architecture of server platform

The diagram of client platform is shown in Figure 5. We provide user a graphical user interface for display of not only cutting face but also map of cutting face and other control inputs such as requests for bookmarks or commands of storing annotations. As a result, in application layer, we implement a "User interface" module that contains three sub-modules.

One is “Mini map” module which is in charge of updating the map of cutting face when orientation or location changes. The second is “Cutting face display” module which is in charge of updating cutting face image sent from server. The third is “Bookmark and annotation” module which handles input commands of storing and retrieving for bookmarks or annotations and send it to server, and then displays the corresponding results such as bookmark list sent from server. Another application module, “Localization” module, manages localization events when user wants to locate or relocate cutting faces. According to different localization methods, “Localization” module will acquire different formats of location inputs from middleware layer or hardware layer and convert it into one mutual format, and then send it to server for location computing. The “QRcode recognition” module in middleware layer is only used when user chose barcode-based localization method to locate cutting faces and it is in charge of decoding the quick response code image captured by camera and then forward decoded location information to localization module. Another middleware module, “Orientation” module, is responsible for computing new rotation matrix when orientation changes and pass it to “Mini map” module to update the map of cutting face. In the meanwhile, “Orientation” module will sent the rotation matrix to server for acquiring new cutting face. In hardware layer, the combination of triaxial accelerometer and magnetic sensor can detect changes of orientation and pass the orientation angles of handheld device to upper layer for further process. The “Wifi” module builds the connection with server at start-up of client program and is responsible for sending requests to server and receiving corresponding response from server. The “Touch screen” module senses user’s touch and forward the touch coordinates to different upper layer modules according to different purposes. The last hardware layer module, “Camera” module, is used when user wants to capture QRcode images.

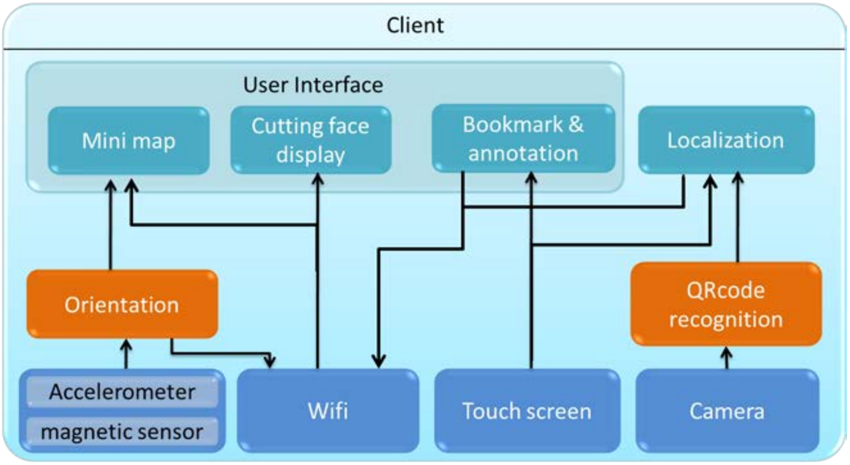

Figure 5: Software architecture of client platform

In next section, we are going to explore the detail of methods employed by modules.

3.4 System Design

3.4.1 Rendering Cutting Faces of 3D object

3.4.1.1 Overview

In our system, we are not doing traditional 3D rendering that converts 3D models to 2D images; instead, we propose a special rendering method to construct the cutting faces of 3D image. In the section of “Cutting Face Construction Method”, we will give definition of a cutting face and introduce our crucial rendering method executed by rendering cutting face module. Besides, according to our scenario, user can see cutting faces with different orientation angles by putting handheld devices in different geographical orientations, but there exist orientation issue, so we proposed two orientation control methods to solve it. The orientation control methods will be introduced in section "Orientation of Cutting Face Control Methods".

3.4.1.2 Cutting Face Construction Method

For clarity of cutting face construction, first, we give the definition of a cutting face. A cutting face S_i is constructed by 2 vectors $\overrightarrow{u_i}$ and $\overrightarrow{v_i}$ and their start point $\overrightarrow{o_i}$ where $\overrightarrow{u_i}^T \overrightarrow{v_i} = 0$, $|\overrightarrow{u_i}| = 1$, $|\overrightarrow{v_i}| = 1$ and $\overrightarrow{o_i}$ is the centre of the cutting face which also represent the location of the cutting face. Besides, the observer's optical direction is orthogonal to both $\overrightarrow{u_i}$ and $\overrightarrow{v_i}$. As shown in Figure 6, continuing using the 3D human head as example, S_1 and S_2 are cutting faces with different orientations and locations. Second, we give the definition of the 3D object. The 3D object is composed of pixels, so we consider it as a pixel matrix, termed $\mathrm{Set}(O)$, and we define that the 3D object is in earth coordinate system; therefore, each pixel of the 3D object has its exact location $\vec{p}$ in earth coordinates which is represented in a row vector, i.e. $\vec{p} = [x\ y\ z]$.

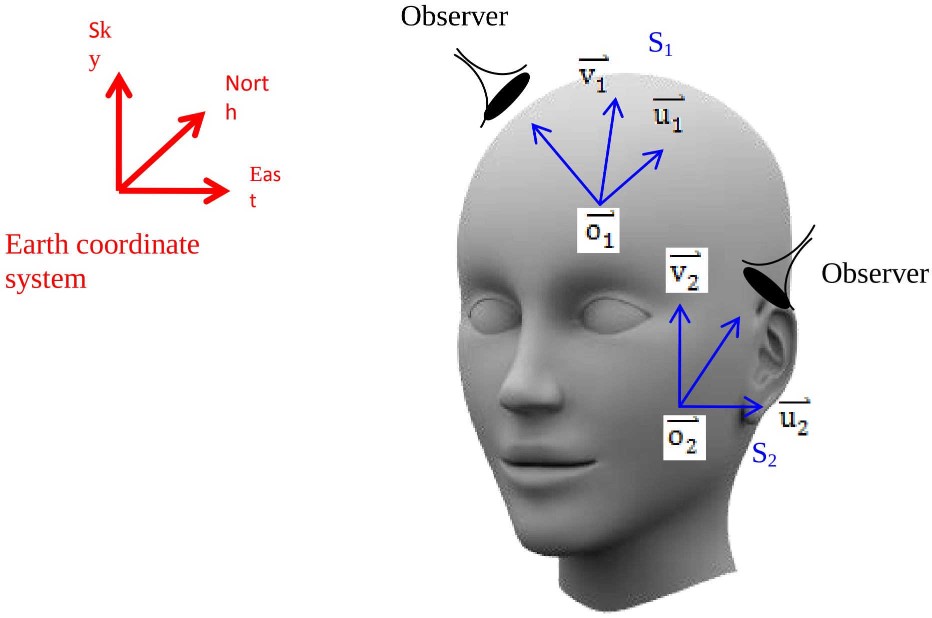

Figure 6: Cutting faces

For cutting faces, we are required to find all pixels' location of the cutting face in 3D object cube to construct the 2D image. To achieve this goal, we can calculate each pixel's location $\vec{p}$ of the cutting face by a simple formula:

$$\vec{p} = \overrightarrow{o_i} + k_u \overrightarrow{u_i} + k_v \overrightarrow{v_i}, \forall \vec{p} \in \mathrm{Set}(O) \quad (1)$$

where k_u and k_v are integer scale factors which are responsible for controlling and identifying which pixel of the cutting face is currently calculated. k_u and k_v have upper bounds and lower bounds, $-\frac{W}{2} \leq k_u \leq \frac{W}{2}$ and $-\frac{H}{2} \leq k_v \leq \frac{H}{2}$,where H represents the pixel height of the image and W represents the pixel width of the image, as shown in Figure 7.

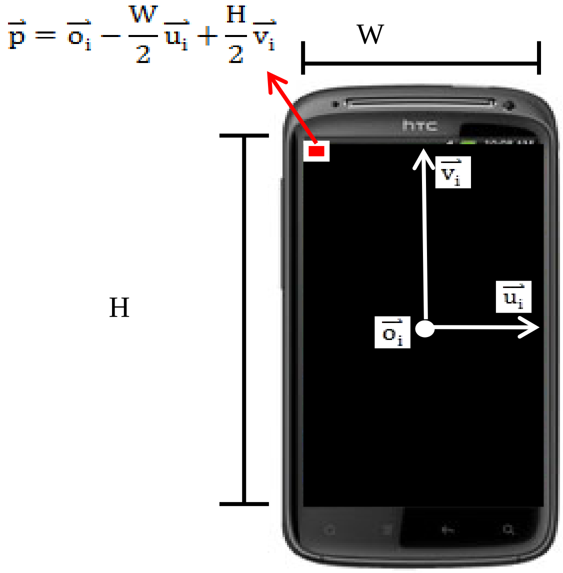

Figure 7: Example of calculating pixel's location

3.4.1.3 Orientation of Cutting Face Control Methods

In out design, user can see any cutting face in any orientation by putting handheld device in user-interested geographical orientation. Therefore, users are able to see cutting faces in 360 degrees by rotating handheld devices about any axis, but if we consider the limitation of hardware ability that handheld devices do not have monitor in back face, we can easily find out that seeing cutting faces in 360 degrees without moving our head or body is not possible. Hence, we designed two control modes to solve this problem and users can switch between these two modes according to their preference. The first mode is "Absolute orientation mode"; the orientation angles of cutting faces follow the geographic orientations of handheld devices. The geographic orientations are detected by combining an embedded magnetic sensor and a triaxial accelerometer sensor, so users can simply rotate handheld devices to preferred orientation, as shown in Figure 8. The second mode is "Relative orientation mode"; users can rotate handheld devices along x-axis or y-axis of handheld devices, and once the rotation and the rotated axis are detected by the embedded triaxial accelerometer sensor, the orientation angle will slowly increase or decrease by δ degrees along rotated axis depending on rotation direction, as shown in Figure 8. The advantage of this mode is that when user finds the monitor's viewing angle is not comfortable, user can switch from "Absolute orientation mode" to "Relative orientation mode" instead of moving his/her head or body to adapt to the monitor.

The means to obtain $\vec{u}$ and $\vec{v}$ defined in previous section depends on orientation modes. In absolute orientation mode, embedded sensors including a magnetic sensor and a triaxial accelerometer sensor will give rotation angles α(roll), β(pitch), γ(yaw) when the sensors detect that the handheld devices are rotated. Then, the orientation module will compute a rotation matrix according to the information of rotation angles. The rotation matrix is listed below:

$$R_{xyz}(\alpha,\beta,\gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha) = \begin{bmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix} \quad (2)$$

The first column is the transposed vector of $\vec{u}$ and the second column is the transposed vector of $\vec{v}$. Differently, in relative orientation mode, the embedded triaxial accelerometer sensor will give gravity values in three axes, and we can analyze the measured value to obtain the information about the axis which the handheld device is rotating about. Then, we increase or decrease the rotation angle of cutting face manually by δ degrees per time unit along the detected rotation axis. Below is the formula of computing the rotation matrix:

$$R^{t}{}_{xyz} = R^{t-1}{}_{xyz} R_d \,,\; R_d = \{R_{xyz}(\pm\delta, 0, 0) \mid R_{xyz}(0, \pm\delta, 0) \mid R_{xyz}(0, 0, \pm\delta)\} \qquad (3)$$

Where t represents time and R_d is the rotation matrix that rotate cutting face by δ degrees per time unit. The first column of $R^{t}{}_{xyz}$ is the transposed vector of $\vec{u}$ and the second column of $R^{t}{}_{xyz}$ is the transposed vector of $\vec{v}$.

In contrast with obtaining $\vec{u}$ and $\vec{v}$, the means of obtaining $\vec{o}$ are the same in two modes. We provide two localization methods which will be introduced in next section, to obtain $\vec{o}$.

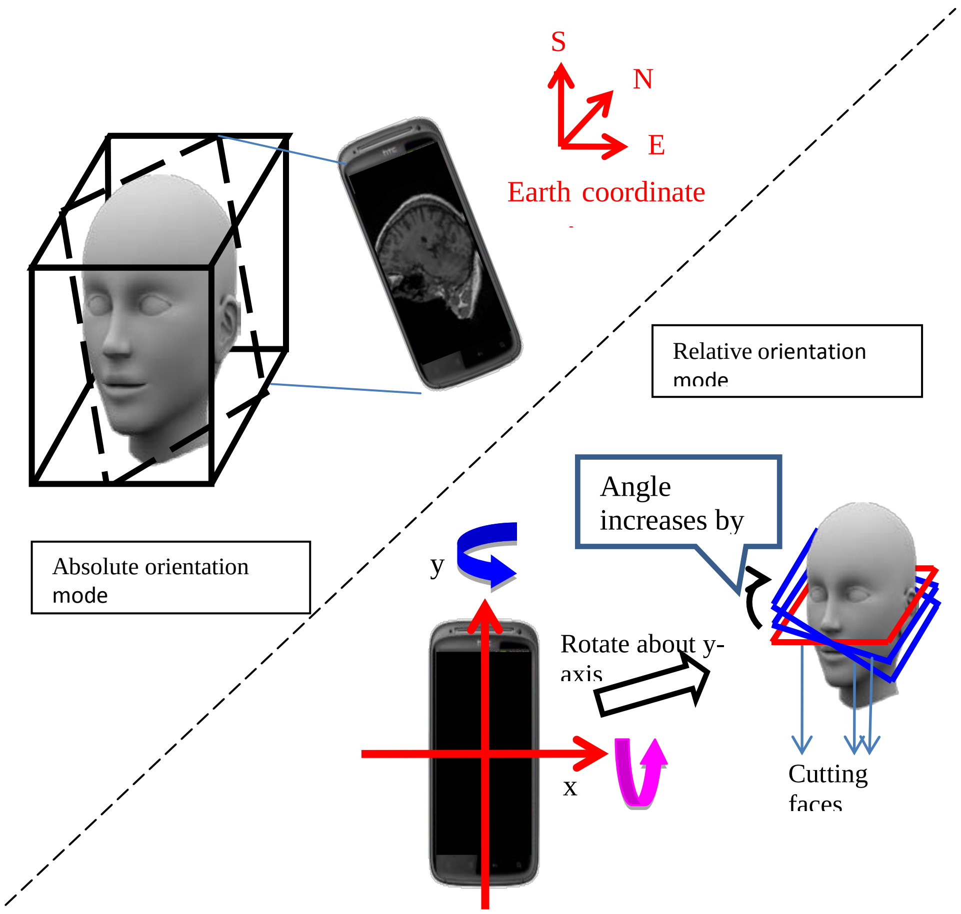

Figure 8: Orientation control modes

3.4.2 Locating Cutting Face

The methods introduced above are all about how to rendering cutting faces, but location of cutting faces are the same important as rendering of cutting faces. Therefore, we provide two localization methods, touch-based localization method and barcode-based localization method, to locate cutting faces and one relocation method to relocate cutting faces after applying localization methods. All methods introduced in this section are implemented by cooperation of localization modules in both client and server sides.

3.4.2.1 Touch-Based Localization Method

Basically, user just has to touch the location he/she wants for the new cutting face on one handheld device such as a tablet, and the new cutting face will show on the screen of another handheld device such as a smart phone, as shown in Figure 9. However, in more than two handheld devices environment, there exists some identification issues. When we have more than two handheld devices in our environment, we have to identify the locating handheld device which will show the new cutting face and the be-located handheld device which is responsible for providing location. Therefore, we provide an identification scheme to identify and pair these two handheld devices. We use a figure to illustrate operations of touch-based localization method with identification phase. In Figure 10, we have a server which is in charge of constructing cutting faces and one tablet, d_1, for location selection and one smart phone, d_2, for showing cutting face with location obtained from tablet. First, user touches d_1 to select and send the new location to server. At the meantime, user touches the synchronization button on d_2 to send synchronization signal to server. Then, server will record timestamps when it receives the location and synchronization signal. After that, server will compare the timestamps to see whether they are synchronous, and if they are synchronous, computing server will construct the new cutting face with the new location and send it back to the smart phone.

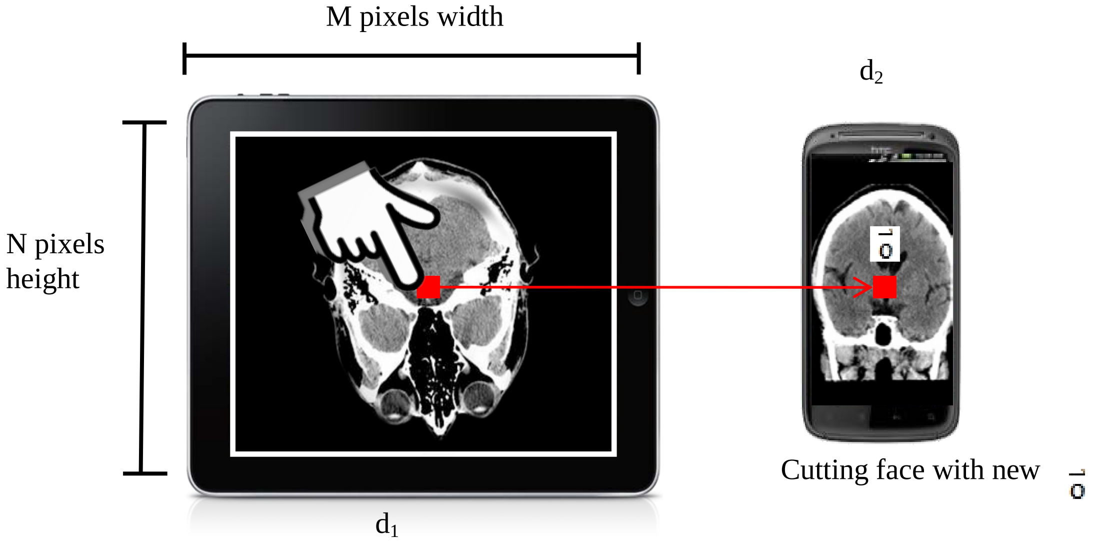

Figure 9: Original touch-based localization method

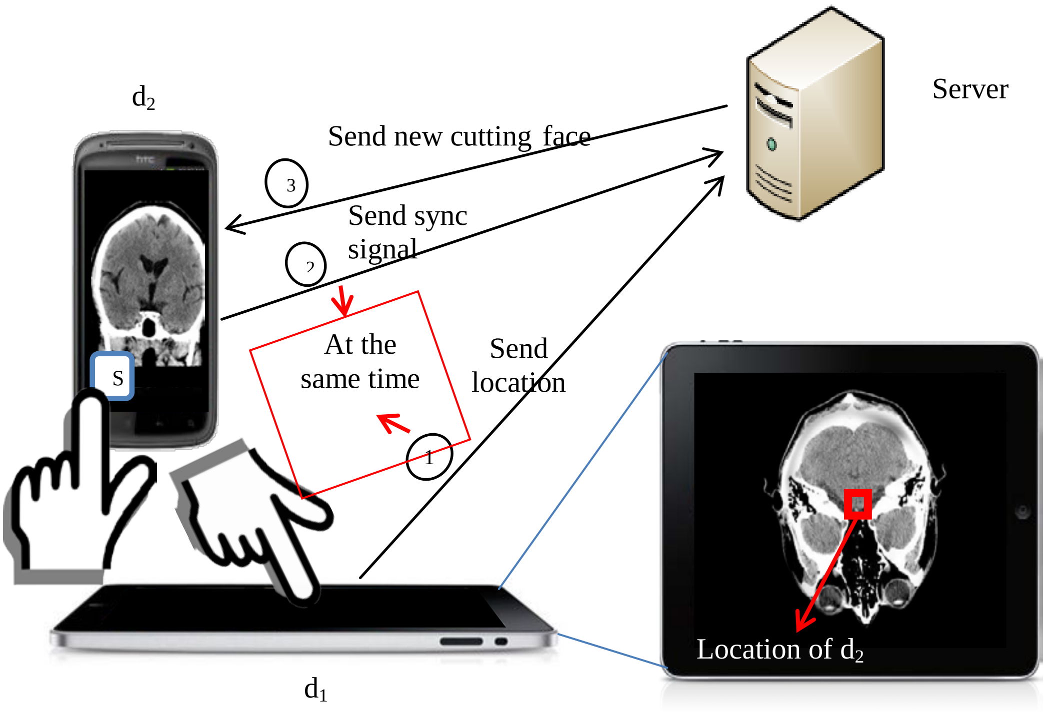

Figure 10: Touch-based localization method include identification phase

3.4.2.2 Barcode-Based Localization Method

Barcode-based localization method is designed for the condition that user have handheld devices in both hands, so touch-based localization method is not convenient for user. Considering this situation, we provide a method that utilizes image recognition technique to recognize the location by capturing image on screen. At beginning, we want to do pixel level localization as touch-based localization method does, but due to limitation of technology that the processing time of arbitrary image recognition without geometric pattern matching is too long, we finally determined to employ the 2D code, quick response code, as our solution. Quick response code has three main advantages. The first advantage is that it can directly encode any literal information like location information inside codes so that we can save the time of doing additional data access. The second advantage is that decoding quick response codes only take little time that is almost a real-time action. The last advantage is that we can generate any size of quick response code according to our preference. The third advantage allows us to generate small quick response code, so that we can use more codes to represent locations on screen, and this make accuracy of location is more acceptable.

The main idea of barcode-based localization method is that we will show quick response codes on screen, and each quick response code represents one location. Then, user can use camera to capture the quick response codes on screen, and by employing image recognition technique, we can get locations after decoding quick response codes. However, barcode-based localization method also has identification issue as touch-based localization method does. As a result, we provide a barcode version identification scheme to solve this problem, as shown in Figure 11. In the figure, we have a server that is in charge of constructing cutting faces and one tablet, d_1, for location selection and one smart phone, d_2, for showing cutting face with location obtained from tablet. First, before localization phase, user should use camera on d_2 to capture ID quick response code on screen of d_1. Then, d_2 will send the decoded ID to server for pairing. Server will pair d_1 and d_2. After that, each time when user wants to get location by using d_2, our system will know the location is supposed to come from d_1.

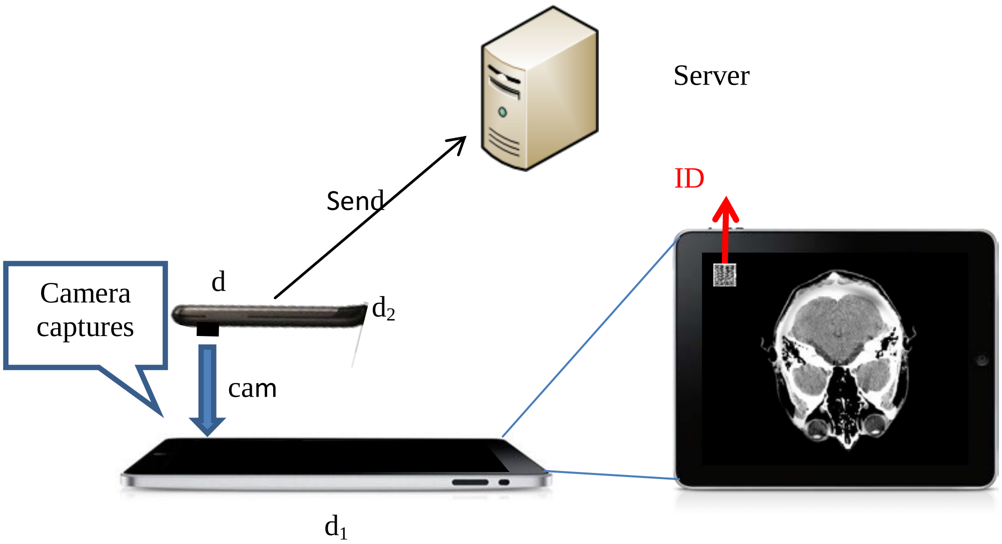

Figure 11: Barcode version identification scheme

About localization, we use another figure to illustrate the detail of localization operations. As shown in Figure 12, first, d_2 will send a start signal to server, and then server will forward the start signal to d1 for noticing d1 to open the quick response code map. Then, user can use camera on d_1 to capture the quick response code on interested location and sends it to server. After getting location, server can construct new cutting face with the new location and sends it back to d_2, and at the meantime, server will send a finish signal to notice d_1 to close the quick response code map. Then, a complete localization is accomplished.

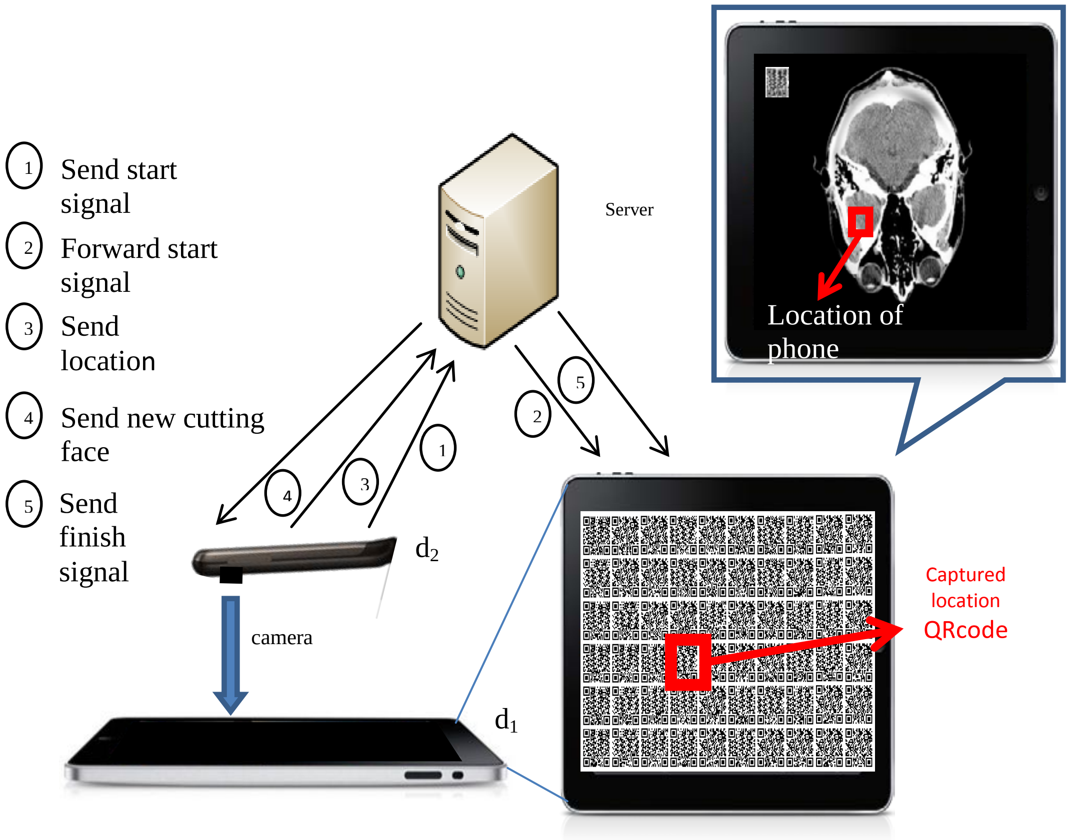

Figure 12: Barcode-based localization scheme

In our system, the location sent from the d2 to the server only represents the location on screen but doesn't represent the location on cutting face. Hence, we need to map the location on screen to location on cutting face. Bellow we will introduce our mapping method. Figure 13 shows the layout of quick response codes on screen. The total number of quick response codes depends on the dimension of the cutting face. We will calculate the ratio of the cutting face's height and width, and then estimate the number of quick response codes. In Figure 13, we assume that the cutting face is M pixel width and N pixel height, and it can afford m quick response codes in a row and n quick response codes in a column. Furthermore, we give each quick response code an ID to distinguish them. The ID is represented by the combination of the quick response code's position in x-axis and y-axis of the map, i.e. $ID = (Q_x, Q_y)$. For example, in Figure 13, the quick response code on right-bottom of the screen has an $ID = (m, n)$. The basic idea of our mapping method is that we overlap the quick response code map and the cutting face, and the pixel point, $P(P_x, P_y)$, which exactly mapped to the center of quick response code is the location that the quick response code represents. The pixel point's location on cutting face, $P(P_x, P_y)$, can be calculated by following equations:

$$P_x = 1 \times \frac{M}{2m} + (Q_x - 1) \times \frac{M}{m} \tag{4}$$

$$P_y = 1 \times \frac{N}{2n} + (Q_y - 1) \times \frac{N}{n} \tag{5}$$

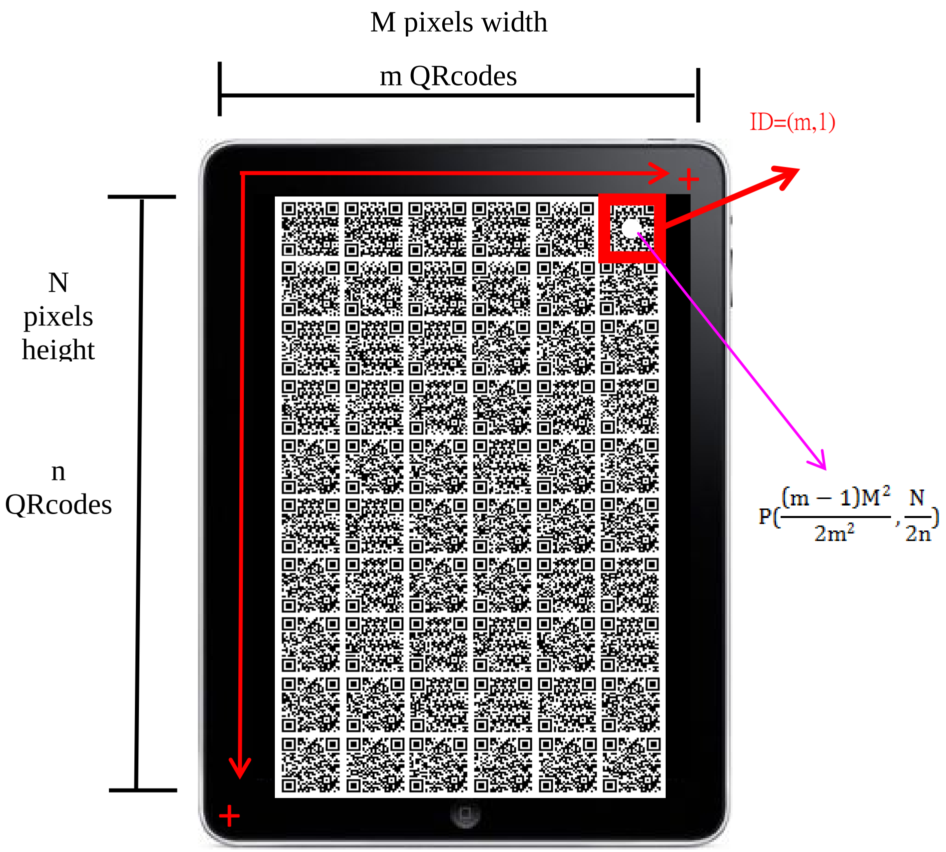

Figure 13: Cascade event

It's worth noting that according to our localization methods' characteristics, we can find an interesting phenomenon called cascade event; the localization events can connect like a chain that one handheld device can get location from another handheld device and offer location to the other handheld device, as shown in Figure 14.

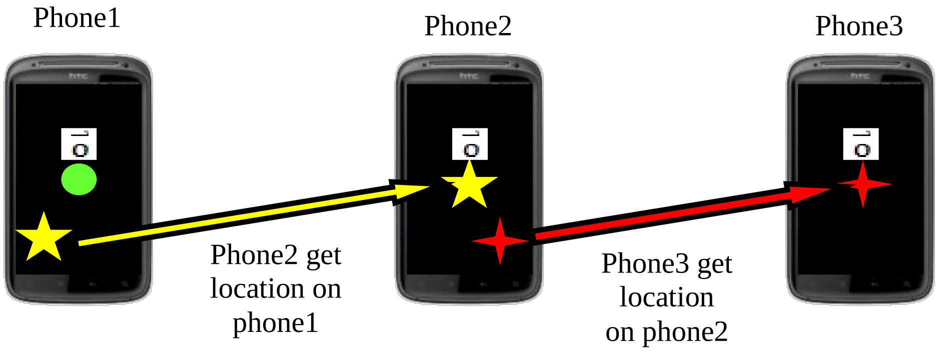

Figure 14: Cascade event

3.4.2.3 Relocation Method

The localization methods mentioned above offer users a way to locate cutting faces in absolute manner. In addition, we provide a relocation method for users to do minus adjustment to location of cutting faces in relative manner. In relocation method, user can use touch gestures to *move* $\vec{o}$, the location of cutting face. We designed two categories of touch gestures to move $\vec{o}$ in different dimensions. One of the categories contains only one gesture called *drag*. Drag is responsible for moving $\vec{o}$ on $\vec{u} - \vec{v}$ plane, as shown in figure 15. $\overrightarrow{v_{drag}}$ is the moving vector which defines the moving direction and the amount of movement. New location of cutting face, $\overrightarrow{o_{new}}$, can be calculate by following equation:

$$\overrightarrow{o_{new}} = \vec{o} + \left|\overrightarrow{v_{dragu}}\right|\vec{u} + \left|\overrightarrow{v_{dragv}}\right|\vec{v} \qquad (6)$$

Where $\overrightarrow{v_{dragu}}$ is a projected vector which is projected from $\overrightarrow{v_{drag}}$ to $\vec{u}$ and $\overrightarrow{v_{dragv}}$ is another projected vector which is projected from $\overrightarrow{v_{drag}}$ to $\vec{v}$.

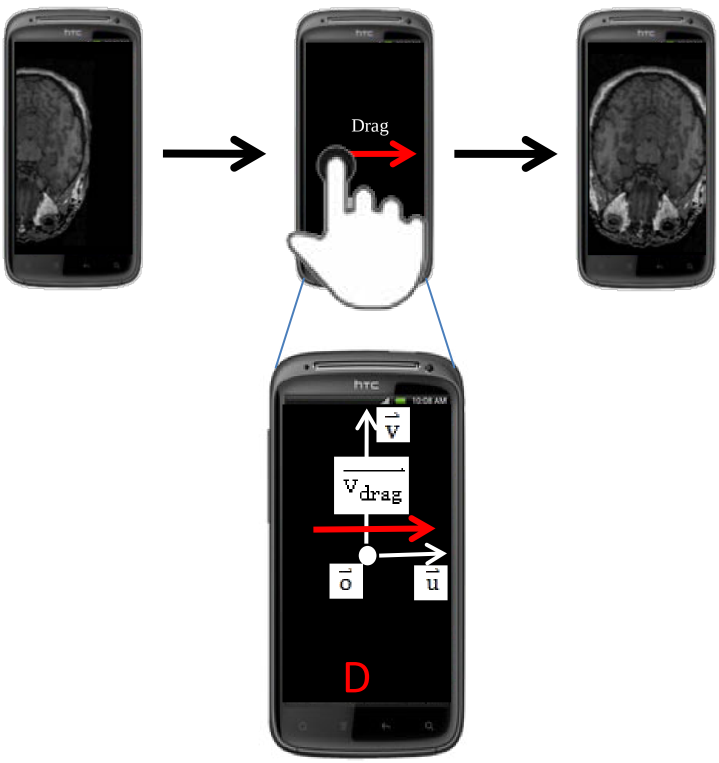

Figure 15: Drag gesture

The other category contains two gestures, *pinch* and *spread*. The gestures in this category are responsible for moving $\vec{o}$ along the axis of $\vec{u} \times \vec{v}$, as shown in Figure 16. Pinch moves $\vec{o}$ in the direction of $\vec{u} \times \vec{v}$ and spread moves in the inverse direction. New location of cutting face, $\overrightarrow{o_{new}}$, can be calculate by following equation:

$$\overrightarrow{o_{new}} = \vec{o} + k_s\vec{w} \tag{7}$$

Where k_s is a scalar factor to control the amount of $\vec{o}$'s movement. k_s is defined as $k_s = d_{new} - d_{old}$. d_{old} is the start distance between two fingers before making gesture, and d_{new} is the finish distance between two fingers after making gesture. $\vec{w}$ is the notation for $\vec{u} \times \vec{v}$.

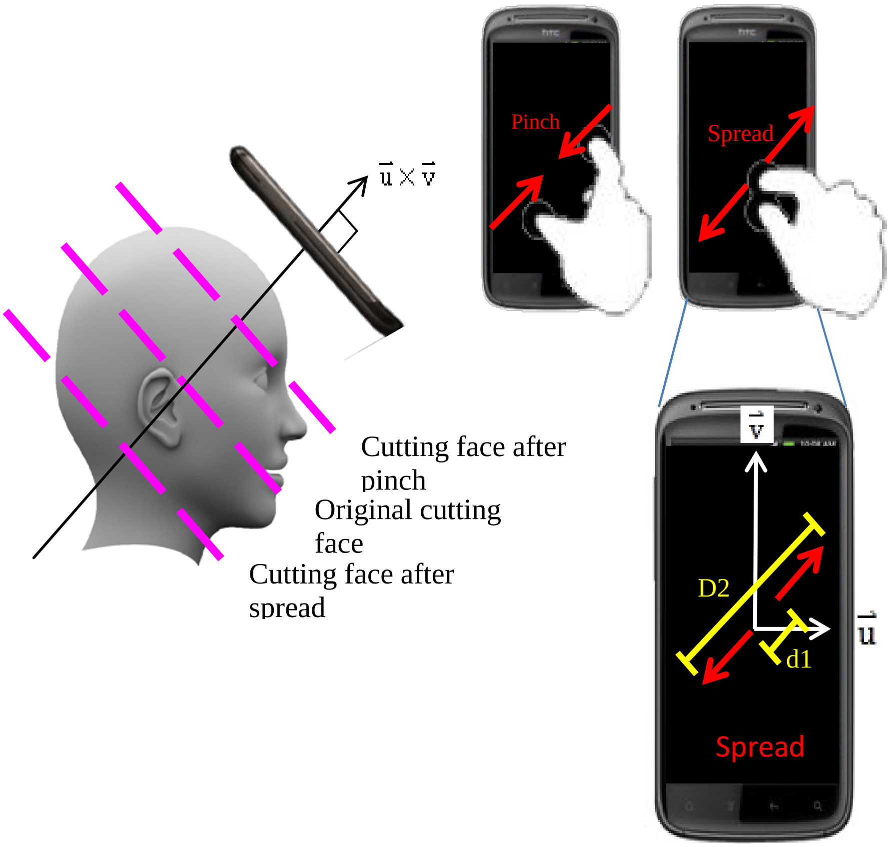

Figure 16: Pinch and spread gestures

3.4.3 Bookmark and Annotation

Considering user experience, it may be difficult for user to re-access a specific cutting face with exactly the same orientation as the cutting face's which she/he has accessed earlier. As a result, we design a function called *bookmark* for users to record any cutting face that she/he thinks it's important or interesting that she/he may re-accesses it the other day. The processes of storing cutting face and retrieving cutting face are shown in Figure 17. If user wants to store the current cutting face as a bookmark, user can send a "Store" signal to server, and then server will record the metadata of cutting face for the use of reconstruction and produce corresponding bookmark information. The bookmark information includes the sequence number of this new bookmark, client id which indicates the owner of the cutting face and the location of the cutting face in 3D image. Then, if user wants to re-access the cutting face, user can send a "Get bookmark list" signal to server to retrieve the information of bookmarks, and then select the desired bookmark to get the cutting face.

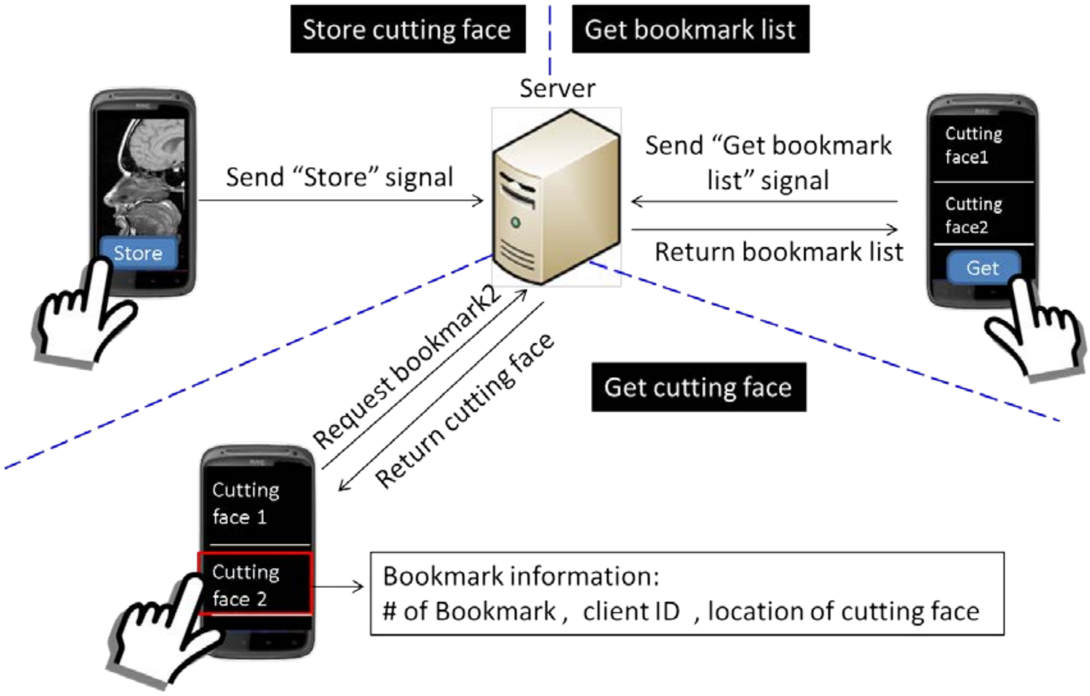

Figure 17: Processes of cutting face storing and retrieving

Furthermore, user may want to put marks on cutting faces as reminders or just write notes on cutting faces to indicate some important information. Therefore, we designed another function, *annotation*, which allows user to annotate directly on cutting faces. For example, if the 3D object is a medical 3D image such as computed tomography image, doctors may want to mark the part of disease, and with annotation function, doctors can easily mark it. As shown in Figure 18, user can touch the screen to annotate on cutting face, and client program will send the drawing points to server. Then, server will store the cutting face with annotation on it as a new bookmark. If user wants to re-access this cutting face, the drawing points will be sent with cutting face to user, and client program will rebuild the annotation by drawing quadratic bézier curve between each two points.

In our system, the information of bookmarks is open to everyone, which means users can access others' bookmarks. Therefore, users can easily share cutting faces or other information through bookmarks. The bookmark and annotation functions are implemented by cooperation of "Bookmark & annotation" modules in both client and server sides.

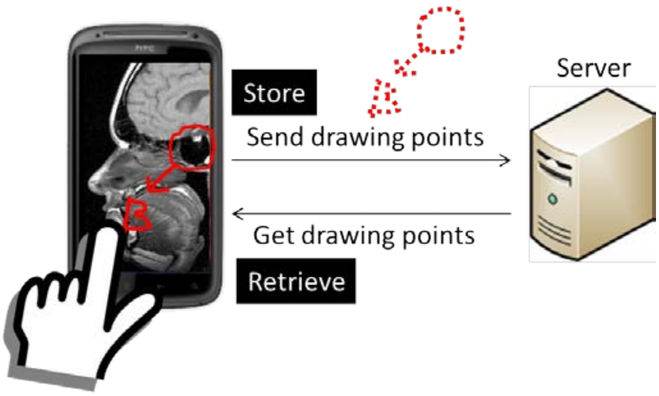

Figure 18: Annotation storing and retrieving

3.5 User Friendly Design — Map of Cutting face

In our system, the 3D object is virtual that user can't see it directly, but by using handheld device to interact with it, user can see cutting faces of the 3D object. However, only seeing cutting faces without other information on screen, user can't have a perfect idea about the positions and orientations of the cutting faces relative to 3D object. Therefore, we provide a small map which will permanently show on screen and real-time update the position and orientation of the cutting face relative to 3D object. As a result, user can easily imagine the relationship between the cutting face and 3D object. Figure 19 gives an example of the map. This function is implemented by mini map module in client side.

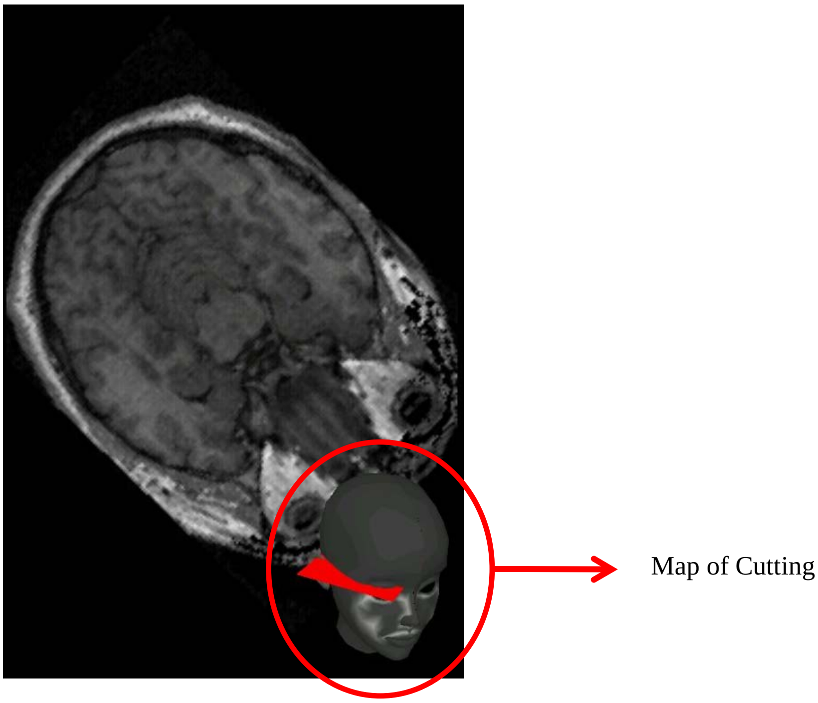

Figure 19: Map of cutting face

4 RESULTS AND DISCUSSIONS- PROOF OF SERVICE

Methodology of Service experience Engineering (SEE) was applied to test the satisfaction of users. We developed a questionnaire for 26 dentists and 30 patients to test the proof of service of our 3D medical image processing systems. Questions of questionnaire are shown in Figure 1. Results of assessments by pair t test indicated that there were a significant increase ($p<0.05$) in both dentists and patients overall experiences (table 1and table 2). Not only dentists gave positive rating and high satisfaction with our new system but also patients' user experience scores increased by 16% after using this system.

Table 1 Pair test-Sample of dentists

	Difference of paired variables					t	Df.	Sig.
	Mean	Standard deviation	standard error of the mean	95% confidence interval				
				Lower bound	Upper bound			
Paired After-Before	5.875	4.673	1.652	1.968	9.782	3.556	5	.009*

Table 2 Pair test-Sample of patients

	Difference of paired variables					t	Df.	Sig.
	Mean	Standard deviation	standard error of the mean	95% confidence interval				
				Lower bound	Upper bound			
Paired After-Before	18.500	5.831	2.062	13.625	23.375	8.947	5	0.00*

The 3D medical image processing through multiple mobile devices can help doctors' consultations and to raise patients' satisfaction by applying SEE (Goldstein, Johnston, Duffy, & Rao, 2002; Hsiao & Yang, 2010). However, some challenges still need to be overcome. First, the cloud computing ability of mobile devices still suffer from the growing big data of medical image processing. Second, the concern of confidentiality and privacy in health has caused the difficulty to integrate with picture achieving and communication system (PACS) and database in hospital information system (HIS). In this study, we can tell that 3D medical image processing systems applied on personal mobile devices would be in a modern need for improving health care. More corrections and modifications based on the vision of cloud services of health care could be made on our new system to improve as Figure 20.

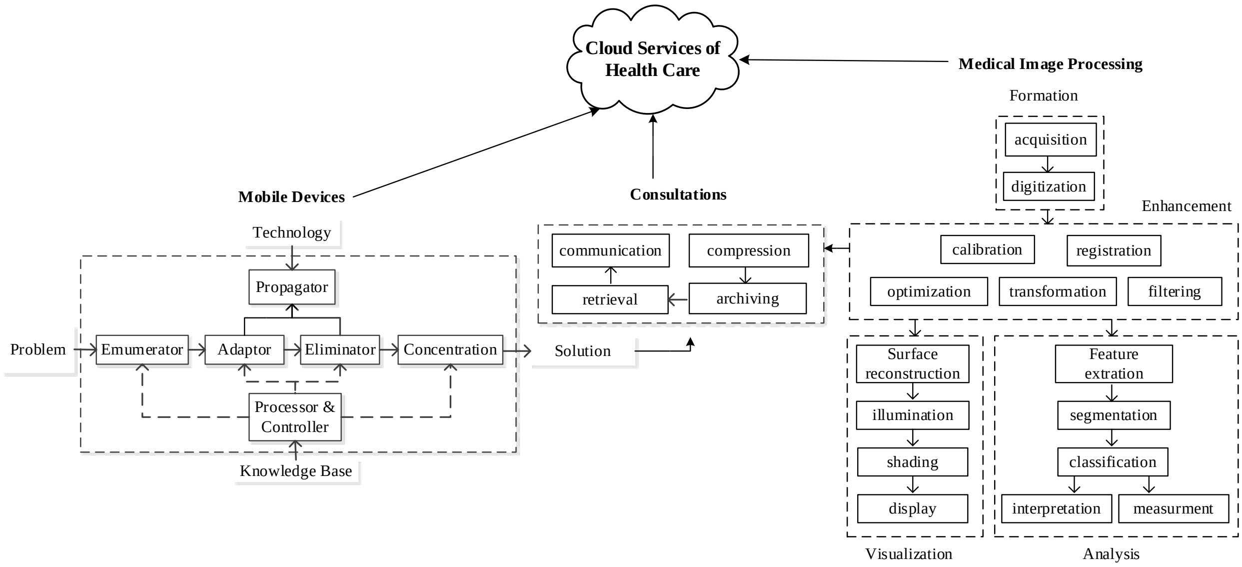

Figure 20: Cloud Services of Medical Image Processing for Health Care

5 CONCLUSIONS

Our prototype combined with cloud service, the first product in concerning field, can automate routine but manual time consuming tasks would benefit doctors and patients. An obvious gap between telemedicine and doctors was closed by this study. Direct apply in the clinical situation could be worthy of expect. SEE was applied to test the satisfaction of users. Results show that both doctors and patients are satisfied with our prototype of 3D medical image processing systems. Conclusions are suggested that the proposed 3D medical image processing systems could be helpful for future health care.

ACKNOWLEDGEMENTS

The author would like to thank the Lab of HSCC at National Chiao Tung University for providing grant support.

BIOGRAPHICAL NOTE

Dr. Peng is a visiting scholar at School of Professional Education and Executive Development (SPEED) of The Hong Kong Polytechnic University (PolyU) teaching courses and conducting research in the field of Health Studies. He obtained a Ph.D. in business administration from National Taipei University (NTPU) and he is also a Ph.D. ABD in the field of computer science at National Chiao-Tung University (NCTU); his interest in applying knowledge of business administration and computer science led him to a focus on interdisciplinary research of health studies. He is an associate professor at Minghsin University of Science and Technology. He was the director of Dept. of Hotel Management from 2005 to 2008. Several opportunities to teach, conduct researches and undergo training in variety countries broaden his horizons. He is now a visiting scholar to teach Award of Health Studies at PolyU SPEED in 2015/2016 academic year. He conducted his visiting research in cloud services for medical image processing at University of Pittsburgh, USA; the research was founded by the Saylin Wen foundation in 2012. He conducted his visiting research in medical image processing at RWTH Aachen University, Germany; the research was founded by the Deutscher Akademischer Austausch Dienst (DAAD) and the Taiwan National Science Council (NSC) in 2011. He was one of Tourism Elites chosen by the Taiwan Tourism Bureau in 2010 to undergo forty days of training in hospitality and tourism in the UK. The Taiwan National Science Council sponsored his 5 months visiting research at University of Strathlyde, UK in 2009. The Ambassador scholarship from the International Rotary Foundation allows him to conduct postdoctoral research for 1 year at Syracuse University, USA from July 2003 to July 2004.

REFERENCES

Butler, A., Izadi, S., & Hodges, S. (2008). *SideSight: multi-"touch" interaction around small devices.* Paper presented at the The 21st Annual ACM Symposium on User interface Software and Technology, Monterey, CA.

Chittaro, L., & Ranon, R. (2007). Web3D technologies in learning, education and training: Motivations, issues, opportunities. *Computers & Education, 49*(1), 3-18.

Francone, J., & Nigay, L. (2011). *Using the user's point of view for interaction on mobile devices.* Paper presented at the IHM '11 23rd French Speaking Conference on Human-Computer Interaction, Antipolis, France.

Geen, J., & Krakauer, D. (2003). New iMEMS® angular rate-sensing gyroscope. *Analog Dialogue, 37*(03), 1-4.

Goldstein, S. M., Johnston, R., Duffy, J., & Rao, J. (2002). The service concept: the missing link in service design research? *Journal of Operations Management, 20*(2), 121-134. doi:10.1016/S0272-6963(01)00090-0

Hansen, T. R., Eriksson, E., & Lykke-Olesen, A. (2006). Mixed interaction space — Expanding the interaction space with mobile devices. In T. McEwan, J. Gulliksen, & D. Benyon (Eds.), *People and Computers XIX — The Bigger Picture* (pp. 365-380). London: Springer

Hinckley, K. P. (2003). USA Patent No.: M. Corporation.

Hsiao, S.-L., & Yang, H.-L. (2010). A Service Experience Engineering(SEE) Method for Developing New Services. *International Journal of Management, 27,* 437-449.

Hürst, W., & Helder, M. (2011). *Mobile 3D graphics and virtual reality interaction.* Paper presented at the The 8th International Conference on Advances in Computer Entertainment Technology, Lisbon, Portugal.

Kray, C., Nesbitt, D., Dawson, J., & Rohs, M. (2010). *User-defined gestures for connecting mobile phones, public displays, and tabletops.* Paper presented at the The 12th International Conference on Human Computer Interaction with Mobile Devices and Services, Lisboa, Portugal.

Lin, C. Y., Peng, K. L., Tsai, J. Y., Tseng, Y. C., Wu, C. C., Chen, M., & Chen, J. (2013). *Apply Cloud Services of Medical Image Processing on Dental Field.* Paper presented at the International Association for Dental Research Meeting, Seattle, USA.

Peng, K. L., Chang, S. K., Tseng, Y. C., & Deserno, T. M. (2012). *Cloud services for medical image processing*. Paper presented at the The 18th International Conference on Distributed Multimedia Systems, Miami, USA.

Rajeeva, K. R., & Sagar, B. M. (2014). Quick response code for fast detection and recognition of information. *International Journal of Emerging Technology and Advanced Engineering, 4*(2), 271-275.

Rekimoto, J., & Saitoh, M. (1999). *Augmented surfaces: a spatially continuous work space for hybrid computing environments.* Paper presented at the SIGCHI conference on Human Factors in Computing Systems, Pittsburgh, USA.

Stavness, I., Lam, B., & Fels, S. (2010). *pCubee: a perspective-corrected handheld cubic display*. Paper presented at the The SIGCHI Conference on Human Factors in Computing Systems, Atlanta, USA.

CPCE Disclosure Authorized