Pers Ubiquit Comput (2010) 14:83–94
DOI 10.1007/s00779-009-0249-0

ORIGINAL ARTICLE

Interaction with large ubiquitous displays using camera-equipped mobile phones

Seokhee Jeon · Jane Hwang · Gerard J. Kim · Mark Billinghurst

Received: 20 November 2007 / Accepted: 7 July 2009 / Published online: 1 August 2009

Abstract In the ubiquitous computing environment, people will interact with everyday objects (or computers embedded in them) in ways different from the usual and familiar desktop user interface. One such typical situation is interacting with applications through large displays such as televisions, mirror displays, and public kiosks. With these applications, the use of the usual keyboard and mouse input is not usually viable (for practical reasons). In this setting, the mobile phone has emerged as an excellent device for novel interaction. This article introduces user interaction techniques using a camera-equipped hand-held device such as a mobile phone or a PDA for large shared displays. In particular, we consider two specific but typical situations (1) sharing the display from a distance and (2) interacting with a touch screen display at a close distance. Using two basic computer vision techniques, motion flow and marker recognition, we show how a camera-equipped hand-held device can effectively be used to replace a mouse and share, select, and manipulate 2D and 3D objects, and navigate within the environment presented through the large display.

S. Jeon
Department of Computer Science and Engineering,
POSTECH, Pohang, Korea

J. Hwang
Image and Media Research Center,
Korea Institute of Science and Technology, Seoul, Korea

G. J. Kim (✉)
College of Information and Communication,
Korea University, Seoul, Korea
e-mail: gjkim@korea.ac.kr

M. Billinghurst
Human Interface Technology Laboratory NZ,
University of Canterbury, Christchurch, New Zealand

Keywords Interaction · Motion flow · Marker recognition · Interaction techniques · Cell/mobile phones · Large display

1 Introduction

The goal of ubiquitous computing is to make computers invisible [1]. That is, people will interact with smart devices or objects in everyday life conveniently and naturally, without recognizing the presence of computers and without significant cognitive effort. Consequently, non-traditional and specialized interfaces will be employed for different ubiquitous computing scenarios. One popular scenario is interacting with applications through large displays such as televisions, mirror displays, tabletop displays, and public kiosks. So far, large displays have been used mostly for one way communication, but in the future, they could be made more interactive and sharable among multiple users. For instance, a "mirror display in a bathroom" has been depicted many times as one of the prominent ubiquitous computing applications [2, 3]. In this scenario, a family member looking in the mirror is recognized and presented with user specific information, such as news, appointments, or to-do lists, and is also able to interact in some way (e.g. turning the TV channel, selection of to-do list, etc.). Such applications are shared among several people, and require only simple selection, manipulation or navigation with minimal, if any, alphanumeric input. Thus, the usual keyboard and mouse input is neither appropriate nor necessary for these simple interactions and also discouraged by the very goal of ubiquitous computing namely, the "invisible" interface.

There are a number of alternatives to traditional input devices for interacting with large displays. One obvious

candidate is the mobile phone. Recently, mobile phones have dramatically improved in terms of their computational power (nearing 1 GHz processing), and are increasingly equipped with various sensing devices like a camera, microphone, and even acceleration sensors. They can also share computational loads with main servers using their improved communication modules such as Bluetooth, wireless LAN, infrared communication, and UWB (ultra wide band technology that supports more than 600 Mbps wireless bandwidth). In the context of association with public devices, mobile phones have the added advantage to provide "private" information and sensory display (e.g. LCD, sound, vibration).

This article introduces user interaction techniques using a camera-equipped hand-held device such as a mobile phone or a PDA for large shared display environments. The camera is used to implement continuous tracking of the mobile device, and buttons for discrete inputs/commands. Using two basic computer vision techniques, the motion flow and marker recognition, we show how a camera-equipped hand-held device can be used effectively to replace a mouse and share, select or manipulate 2D and 3D objects and navigate within the environment presented through the large display. In particular, we consider two different cases, sharing the display from a distance, and using the mobile device as a second input device for interacting with a touch screen display.

In the next section, we review other researches that are related to this work such as interaction using hand-held devices and vision-based tracking. Then, we present three scenarios as starting points for interaction design for the two target usage situations. Based on the task analysis and requirements, Sects. 4 and 5 describe the specific proposed interfaces and their implementations. Next, we report and discuss our experiences in using the proposed interfaces. Finally, we conclude the article with a summary and plans for future work.

2 Related work

Possibilities for interaction with smart hand-held devices (e.g. palmtop computers) were first investigated in 1993 by Fitzmaurice et al. [4]. In this article, the authors suggested several methods for display and 3D interaction using palmtop computers in a virtual reality (VR) application. Watsen et al. have also used a PDA to interact in the virtual environment, but this interaction was mostly button or touch screen based and no PDA tracking was used [5]. Kukimoto et al. also developed a similar PDA-based interaction device for VR, but with a 6DOF tracker attached to it. Using this they were able to demonstrate 3D interaction such as 3D drawing with the PDA (moving it and pressing the button or touch screen) [6]. Mantyla et al. used an accelerometer for detecting user's hand gestures (with Hidden Markov Models) for interaction with hand-held devices [7]. Accelerometers give better performances on movement-oriented gestures; however, many gestures are position sensitive and so are better handled with a vision-based approach.

There were several previous researchers who tried to use the camera as an interface for user interactions, especially for 3D interactions [8, 9]. However, robust user's motion tracking is not an easy problem, particularly in unconstrained environments and with the limited computational power of mobile phones [10]. As mobile phones become more powerful, they are becoming more useful as a platform for computer vision. For example, Wagner et al. implemented a marker-based augmented reality (AR) system using a self-contained hand-held device and applied it to their textual environment augmentation project [11]. Paelke et al. developed a hand-held AR-based soccer game which uses a camera for detecting kicking gestures [12]. Hachet et al. used a camera-equipped hand-held device as a prop for interaction in a virtual environment. In their system, the hand-held camera recognized the movement and poses of a special marker held in the other hand [9, 13]. The user could interact in the virtual environment (seen through a separate large display in front of him) by detecting the motion of the marker by the camera. Hansen et al. used a camera-equipped mobile device to establish a spatial relationship between a virtual environment and the physical space to form a mixed reality space [14].

There have already been previous attempts to use camera-equipped phones for interaction with large display systems. Such a combination has already been emerged as a popular system configuration according to Kruppa et al. [15]. Ballagas et al. used camera-equipped mobile phones for interacting with the large vertical displays. They used the optical flow algorithm and software markers for tracking the mobile device [16]. They also considered the problem of multi-user collaboration in a large shared display using mobile phones. Their work was partly based on a survey they conducted on the use of a mobile phone as a ubiquitous input device. In the survey, they categorized interaction methods according to the interaction dimensionality (1D/2D/3D), task (position/orientation/selection), continuity (continuous, discrete), and directness (direct/indirect), and noted that more attempts were needed in applying mobile phones to "3D" interaction. Rohs et al. too combined mobile phones with the large public display. Marker-free user motion tracking was achieved using a similar optical flow algorithm and software markers on the display [17]. The Spotcode project also focused on the usages of the mobile phone in the large shared display. They designed context-related markers for the tracking

[18]. Special devices such as infrared LEDs also have been used to help detect the position of the hand-held device and for interaction [19]. Our work builds on these lines of researches in terms of proposing a new way of interacting with large displays using camera-equipped phones.

The most popular forms of public or sharable displays are relatively small (sharable by only two or three) and thus can be implemented satisfactorily (in terms of interaction requirements) with touch screens [20, 21]. Note that most current touch screens can only support one touch at a time, thus are not appropriate for interaction among many number of people. For larger scaled systems with more complex interaction requirements (which is our target application area), one possible approach is to employ laser pointers (equipped with buttons) with special "spot" detectors behind the display [22]. For comparison, it would be certainly possible to install a laser pointer on a mobile phone to mimic such an approach. However, again it would be difficult to distinguish between laser spots among multiple users. Regenbrecht et al. developed a versatile tabletop interaction system with a sharable display but required special setups including multiple cameras, tabletop projection, and interaction devices (e.g. a commercial digital pen) [23]. In contrast to these efforts, in our work we restricted our system to be as self-contained as possible, that is, to work without any external installation of physical sensors or markers. In addition, our goal is to support multiple users using and sharing a relatively large display.

3 Interaction scenarios and requirements

As a starting point, we present three distinct scenarios for using the camera phone for interacting with a large display. The first scenario involves the task of sharing and exchanging information presented in a distant large display among multiple participants (See Fig. 1; Scenario 1).

Scenario 1

The marketing team is gathered in the presentation room, discussing the market strategy for the next phase and job assignments. Each member of the team points their mobile phone toward the display and uploads his or her idea of what their jobs are. Each person moves one's mobile phone to post their ideas without cluttering the overall display. Bill, the manager, tries to sort through the information, prioritizes and makes an ordered list of assignments. He does this by pointing his mobile phone at the screen and selecting the member's posts, moving and dragging them here and there, and even deleting some of the unnecessary ideas. Using his mobile phone as the interaction device, he creates copies of several arrows and moves, elongates and rotates them between the assignments to draw a big flow diagram of assignments. When everything was agreed, Bill presses a button on his phone to save the document. Gerry selects the modified version of his posts and copies that back into his mobile phone and confirms it through the phone (private) display.

As can be seen in this scenario, large public displays usually do not employ touch screen systems that would otherwise allow direct selection and manipulation of objects. This may be due to cost (having to cover a large area), technology (e.g. multiple touch not possible), operational reasons (e.g. required maintenance) or physical inaccessibility (e.g. not being able to reach top portion of the display). Note that while a simple virtual mouse might suffice for this particular scenario, our ultimate purpose is also to find creative uses for the "display" capabilities of the hand-held devices (as illustrated in the last part of the scenario). The scenario demonstrates the needs for the usual fundamental tasks such as object selection, translation, rotation, and scaling. In our context, one alternative solution is to integrate a wireless optical mouse capability into a mobile phone. This solution is not sufficient because a 2D mouse (1) is not appropriate for contents that require 3D interaction, (2) requires an operating surface that might not be available all the time, and (3) may not support multiple mouse input. Another possibility is the use of the "virtual desktop" mapping the public display to the small mobile phone display (assuming the mobile phone display has a touch screen). However, the size and resolution of the mobile phone display is too small to support effective collaborative interaction as depicted in the preceding scenario.

The second scenario involves a more intimate use of the display from a close distance using a touch screen and cell phone input (See Fig. 2; Scenario 2).

Scenario 2

Jack and Jill are preparing a joint report. Jack pulls up couple of files on the tabletop display and tries to show one to Jill. He chooses the document with his finger on the touch screen and drags it toward Jill who is sitting at a different side of the table, and at the same time rotates his cell phone, with the other hand, to rotate the document toward Jill. Jill finds the document to be too small for her viewing pleasure and enlarges the scale by selecting an anchor point on the document, dragging upward to make space, and gestures her cell phone, with the other hand, side ways to enlarge the document all at the same time. By a touch of a button, the cell phone is turned into a magnifying glass allowing Jill to examine the fine details of the figure in the document (2D Interaction). Jill has a 3D model of a product from her company in her mobile phone. She wants to discuss with Jack about the product. She copies the 3D model file to the tabletop display. She would like to show to Jack a closer look of the bottom part of the model. She rotates the model and zooms into the model using her mobile phone (3D Interaction).

Scenario 2 illustrates the need for two-handed simultaneous interaction for object rotation. Without the two-handed input, the object rotation must be carried out as a sequence of operations. Touch screen systems usually do

Fig. 1 A scenario for using camera phones in a shared environment with a distant display

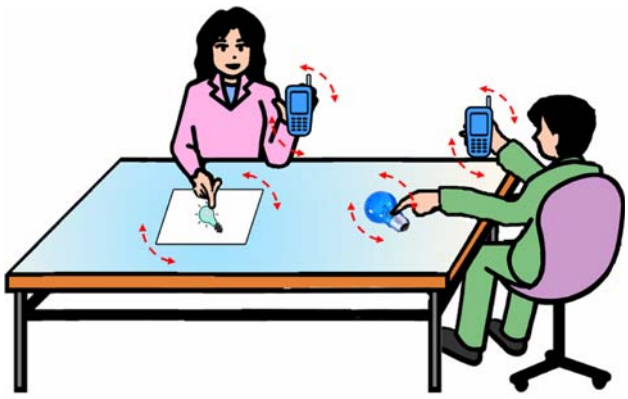

Fig. 2 A scenario for using camera phones in a tabletop display environment with a touch screen

not allow simultaneous two-handed input. Thus, one solution is to use a smart tangible prop, like a sensor-equipped mobile phone, as an embodied interface.

The final scenario illustrates the novel integration of motion gesture-based interface with the private display (see Scenario 3). The metaphoric motion gesture-based interface enabled by the sensors/camera on the mobile phone can enhance the overall experience of the game play (or other location-based contents) [24]. Note that while we expect a complex system command structure (e.g. through a hierarchical menu system) will not be necessary by the nature of highly specialized ubiquitous computing applications, if necessary, it can be implemented by a combination of the sensor enabled continuous tracking and button commands.

Scenario 3

Andrew and Ellen like to play a multi-player racing game on their Bluetooth connected mobile phones. They like it even better when they can seamlessly connect it to their large screen television at home. The large screen TV shows the whole racing track with positions of each player (with other friends or family members cheering on) and incoming obstacles and gift points, while the mobile phone displays show first person views of the on-going racing game. Moreover, instead of button presses, they "rotate" their phones to change direction (like a car handle) and "push and pull" the phones to control acceleration.

4 Proposed camera-equipped phone interfaces for large displays

To summarize the interaction requirements, a ubiquitous computing application with a sizable display will be driven mainly by object selection and manipulation through two subtasks: continuous positioning and making discrete commands. Although not highlighted in the three usage scenarios, navigation or search (e.g. going to the next channel or simple menu/content browsing) is also an important task that can be realized by the same subtask(s) and also by interpreting user motion. In this section, we describe a specific mobile phone-based interface implementation for interacting with 2D/3D environments through a large sharable display: (1) as a 2D/3D "fly" mouse, (2) as an embodied agent, and (3) as a medium for motion gesture. In all cases, continuous tracking is necessary, and this is implemented using two basic computer vision techniques, motion flow, and marker recognition (See Section 5).

The interactive large display system for which the proposed camera phone interaction is to be applied is shown in Fig. 3. The Nokia 6630 mobile phone with the Symbian Series 60 (Second edition FP2) as its operating system [25] is used. To implement the marker-based approaches, we have used the ARToolkit [26] for the Symbian OS. All optical flow calculation and marker tracking algorithm (See Sect. 5) were carried out within the hand-held device (i.e. not on the server side). The communication between the device and large display system (server) was made by a Bluetooth connection. For example, the motion vectors of the tracking features computed with the hand-held device were sent to the server. Likewise, marker transformation matrices were sent to the server when the marker-based approaches were used. The large display system uses (when operated from a close distance) the NextWindow

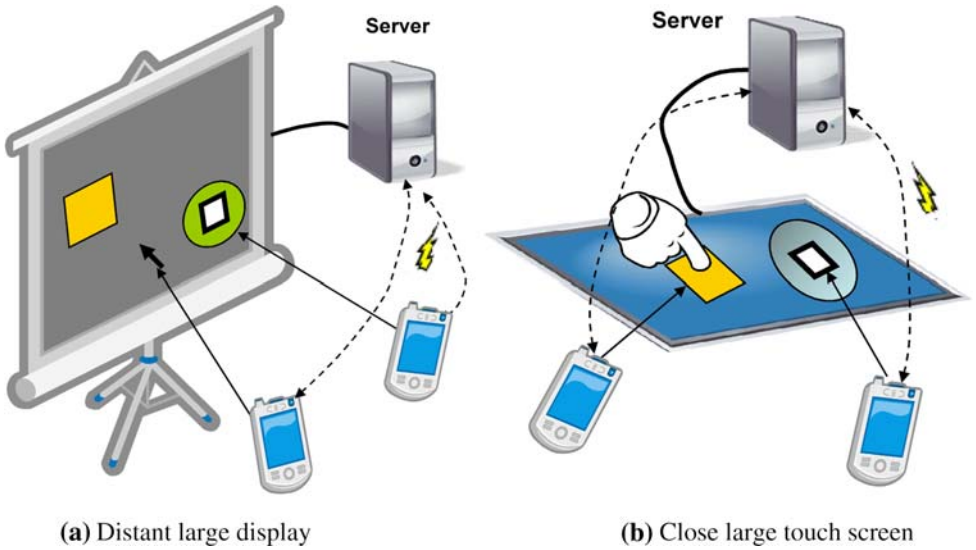

Fig. 3 The overview of the interactive large display system: **a** multiple users interacting with a distant display using cell phones and **b** with a close touch screen display. Tracking data is conveyed wirelessly to the server which manages the application/display

(2100 series touch frame) [27], a touch screen that allows one finger touch input at a time.

4.1 Interacting from distance (without a touch screen)

4.1.1 Motion flow-based approach for 2D interaction

The motion flow-based approach uses a cursor which indicates the position of user's interest. The basic sequence of a selection is identical to our normal use of a 2D cursor. We see a cursor, locate it over the object and select the object right beneath it (using a button). One benefit of this method is that it is not necessary to find out the absolute position of the interaction device because a cursor always indicates the absolute position of the user's interest. Thus, the tracking module only needs to find the relative movement of the interaction device using motion flow-based relative tracking. After selecting an object, we can move the object to a different location by dragging the cursor. The movement of the cursor is obtained from the relative movement vectors of the optical flow [28, 29]. The implementation of the relative motion tracking is described in Section 5.

A usual 2D rotation requires a specification of the rotational axis and continuous tracking for specifying the amount and direction of the rotation. Once an object is selected, the proposed cursor-based rotation assumes a rotational axis (2D point on the object), and maps the amount and direction of rotation (See Fig. 4).

4.1.2 Marker-object approach for 2D/3D interaction

Although the cursor-based interaction (enabled by the motion flow) is quite intuitive to use, it has several shortcomings. The most serious one is that we always have to move a cursor to the appropriate position before selecting the object. This redundant manipulation would be a major performance hurdle if the task was time critical.

In this approach, a marker is assigned to each object. Through marker tracking, we can select an object by just putting the object on the centre of the camera view and pressing selection button on the phone. As soon as we press the button, each object is overlaid by a virtual marker (previously hidden) and by identifying/tracking the marker, the object can be selected. After the object selection, the overlaid markers disappear except for the selected object's marker. Since we know the position of selected object's marker in the camera view projection plane, forcing that marker's position to the centre of the camera viewing plane is easily accomplished. As a result, the selected object moves with the centre of the camera viewing direction. The user merely needs to point the phone cam toward the new position (in moderate speed, See Sect. 5). For rotation, as the marker tracking module computes the relative location and orientation of the marker relative to the camera, we can use this information to rotate the object and align it with the viewing camera. Therefore, when the camera phone is rotated, the object in the display will rotate accordingly (around an assumed z-axis point). The proposed interaction sequences are shown in Fig. 5. The data exchange between the phone and the server can be accomplished via short distance communication, e.g., by Bluetooth or infra-red (mostly available on today's cell phones).

4.1.3 Marker-cursor approach (2D/3D interaction)

In applications such as drawing and writing, a reasonably fine control of a small 2D/3D cursor becomes a major requirement. As seen in Fig. 6, the marker-cursor approach

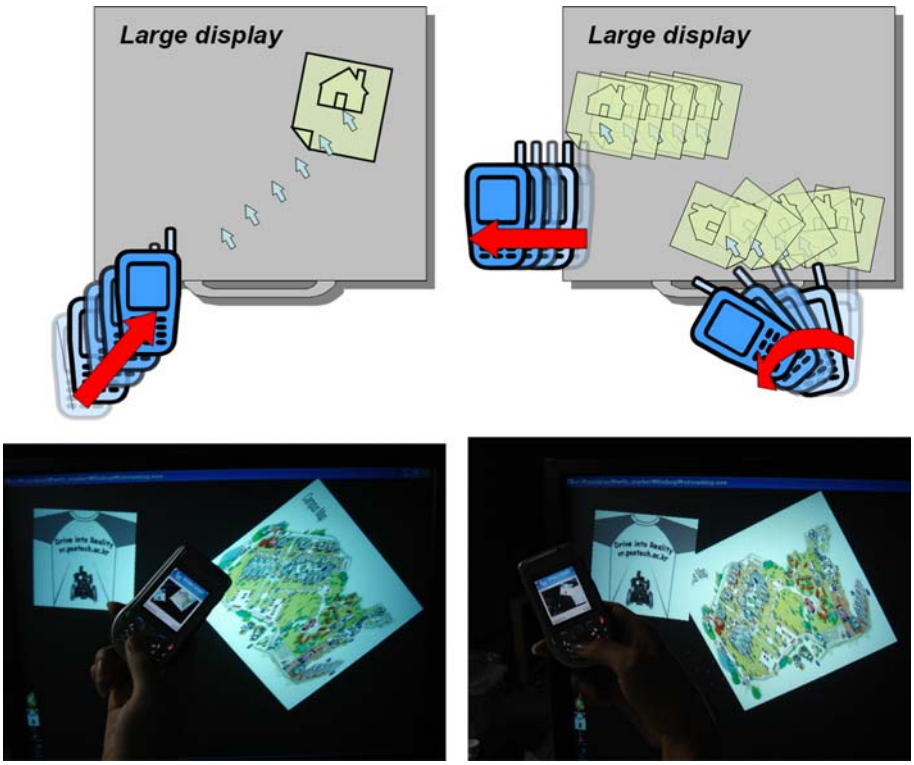

Fig. 4 The upper illustrations show how to select (*left*), translate, and rotate an object (*right*) using a motion flow-based tracking with a camera-equipped mobile phone. The *lower pictures* show the actual implementation results

enables continuous and fine control of the cursor in the large screen environment. Once the camera of the hand-held device detects the marker-cursor displayed on the large screen, we can calculate the 6-DOF information of the hand-held device in the environment. The performance of the marker motion is in general much more robust than using the motion flow approach (See Sect. 5). The intersection point of an orthogonal ray from the hand-held camera and the large screen becomes the position of the marker-cursor. While the camera sees any parts of the screen, the cursor moves to the intersection points of the ray from the camera and screen and this makes continuous and fine movements of the cursor. As shown in the right parts of Fig. 6, the fine control of the cursor such as writing letters becomes feasible with the suggested marker-cursor approach. The 6-DOF motion information of the hand-held

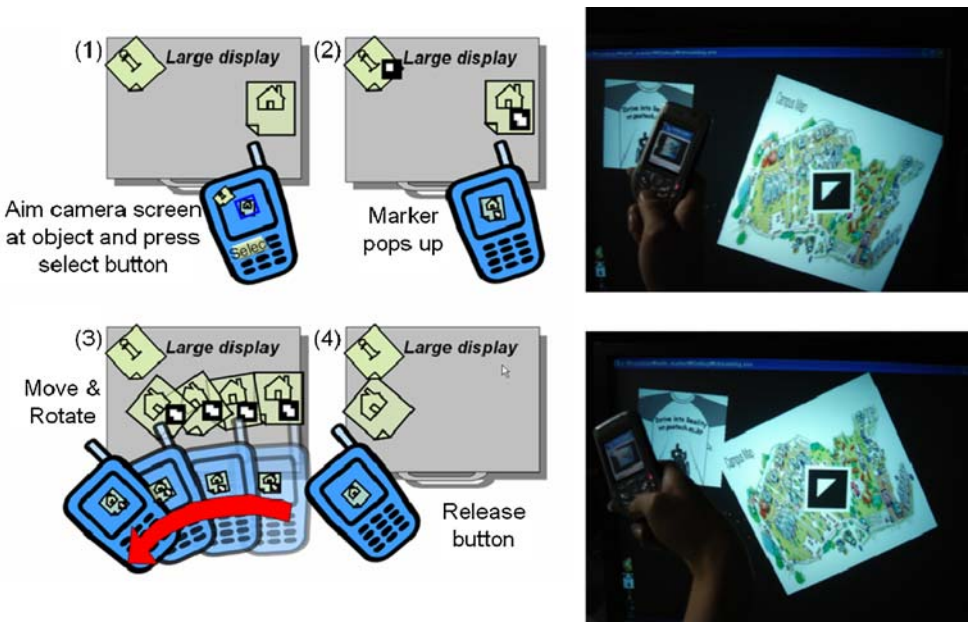

Fig. 5 The left illustrations show an interaction sequence (for 2D rotation) of the marker-object-based approach. The *right pictures* show the actual implementation

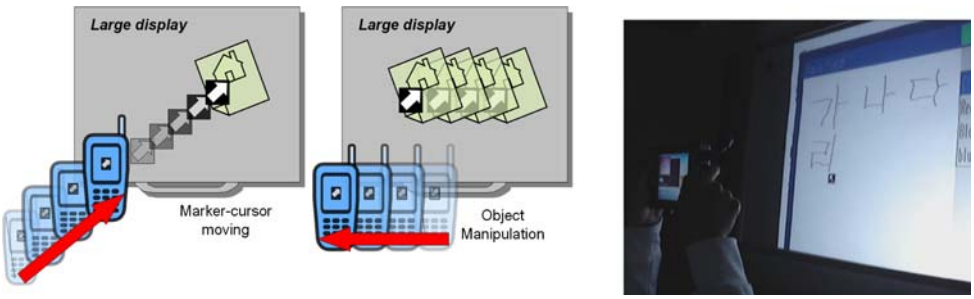

Fig. 6 The *left illustration* shows how to locate and manipulate a cursor in the marker-cursor approach. The picture in the right shows the use of marker-cursor in a writing application

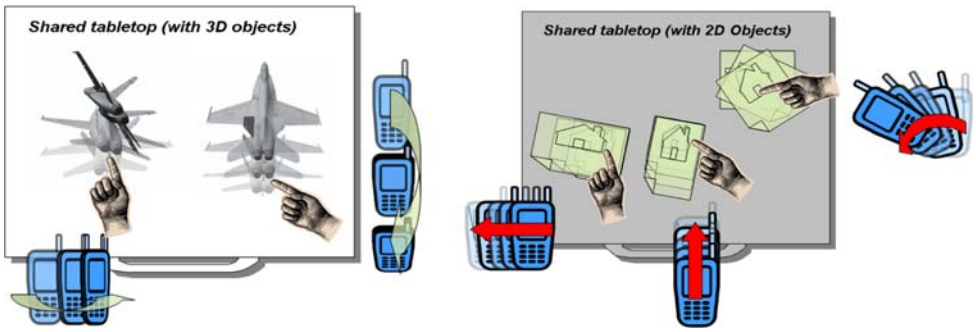

Fig. 7 Object rotation and scaling using the mobile phone and one finger touch. The task is accomplished efficiently in a simultaneous manner for both 3D (*left*) and 2D (*right*) objects

device is computable relative to the marker, so it is also possible to extend this approach to the 3D applications.

4.2 Interacting in proximity to touch screen (mobile phone as an embodied "auxiliary" agent)

As illustrated in the scenario, when the display is to be used in close proximity to the user, a touch screen is probably the most natural and best interaction device. Here, we propose an approach for object rotation using the mobile phone for a two handed simultaneous interaction. Note that most touch screen systems usually do not allow simultaneous two-handed input. Without the two-handed input, the object rotation must be carried out in sequence. The proposed solution can carry out object selection and translation through the touch screen interface, while mapping the mobile phone's (manipulated by the other hand) rotation to the target object's rotation. The mobile phone's rotation is tracked using the motion flow technique so that the user does not have to aim at any special tracking markers. A scaling operation can be carried out in similar manner. This is an example of the mobile phone being embodied as the target interaction object.

Constrained (since not all 6 degrees of freedom can be robustly tracked using the current motion flow technique) 3D object manipulation can be carried out similarly as well. A touch screen is basically a 2D device such as a mouse or a joystick, and previous researches have shown the shortcomings of such 2D-based devices for 3D interaction [30, 31]. The use of typical 3D VR devices such as special trackers is also prohibitive (e.g. cost, availability) for the mass-driven ubiquitous computing environment. Figure 7 illustrates the interaction process, and Fig. 8 shows the actual implementations. In this case, the object pivoted by the finger touch is rotated or scaled according to the rotated camera phone.

4.3 Motion-based gesture interaction

Finally, the raw-tracked motion data (either by motion flow or marker recognition) can further be interpreted for simple abstracted motion gestures. The recent success of the Nintendo WII console [32] and its gesture-based game interface, shows the potential for gesture input to improve the user experience. Figure 9 shows a simple example of imitating a car handle by detecting a rotation pattern in the roll direction, and an acceleration pedal by detecting the forward/backward push. Complex motion gesture recognition is starting to become feasible with the increased CPU power of today's new mobile phones.

5 Relative motion tracking and its performance

5.1 Implementation on the hand-held device

The relative tracking process is divided into two stages: (1) extracting features from two consecutive image frames and establishing correspondences (which gives us the 2D image motion flow) and (2) estimating the motion parameter changes of the camera (i.e. the interaction device). For the motion flow and feature tracking, we have used the

Fig. 8 Snapshots from actual implementations of two-handed interactions using the mobile phone and touch screen. *Top row*: 2D object rotation, *Middle row*: 2D scaling, *Bottom row*: 3D object rotation

Fig. 9 Motion gesture recognition for the car driving metaphor. Roll rotation for direction handling and forward/backward movement for accelerator pedal control. An implementation is done using an UMPC (for simple gestures, a mobile phone implementation is possible, too)

pyramidal Lucas–Kanade optical flow algorithm [28]. This algorithm first applies corner detection for finding feature points and then uses image pyramids for tracking those feature points. Due to the limited of computational power of our mobile phone, we adjusted and varied the load by the desired frame rate, image resolution of the camera, and the number of feature points by trial and error. As a result, we have used 10 feature points with the camera image resolution of 160 by 120 and obtained about 6–7 frame/s tracking rate.

With motion flow on the image alone, it is not possible to decide if translational movements along the x-axis in the image resulted from translational movement of the camera or by rotation of the camera around the y-axis and similarly between translation along y-axis or rotation around x-axis (See Fig. 10 for the local coordinate system on the hand-held device and the screen). Thus the problem is simply solved by using separate interaction modes (i.e. one for translation and another for rotation). In each respective mode, velocities are measured from the motion flow and integrated to compute the relative amount of changes in translation or rotation. Due to the relatively low sampling rate (6–7 frames per second), linear interpolation was employed on the server side for smooth cursor movement on the display.

Recognizing the z-axis rotation (roll) from motion flow is, however, somewhat difficult because the rotation always

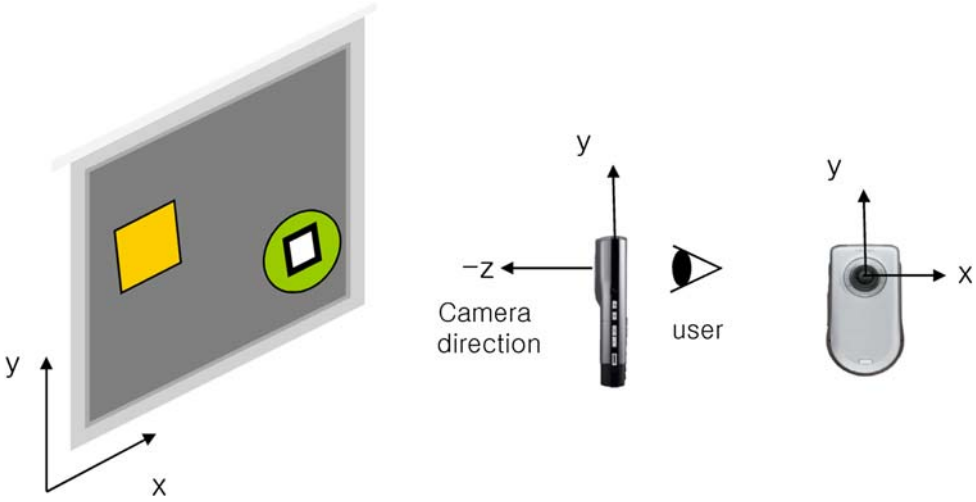

Fig. 10 The local coordinate system on the hand-held device. The z-axis is assumed to be perpendicular to the display surface

usually entails translation and only the z-axis rotation factor must be extracted from the complex feature vectors. Our simple z-axis rotation estimation algorithm is as follows (See Fig. 11). First, since we know the locations of the feature points, we can compute the quadrants they belong to in the image plane. Then we calculate four means of the movement vectors in each quadrant (big blue arrows in Fig. 11). Then, we compare the means of the first quadrant to that of the third, and the second to the fourth. If these differences are higher than some threshold, we recognize it as a rotation in the z-axis, and map its magnitude to the amount of rotation.

In contrast to the motion flow-based algorithm that only provides 3 degrees of freedom tracking (x, y translation and roll rotation) for 2D interaction, the marker-based approach uses the ARToolKit [26] that offers a full 6 degrees of freedom tracking (x, y, z translation and roll, pitch, yaw rotation).

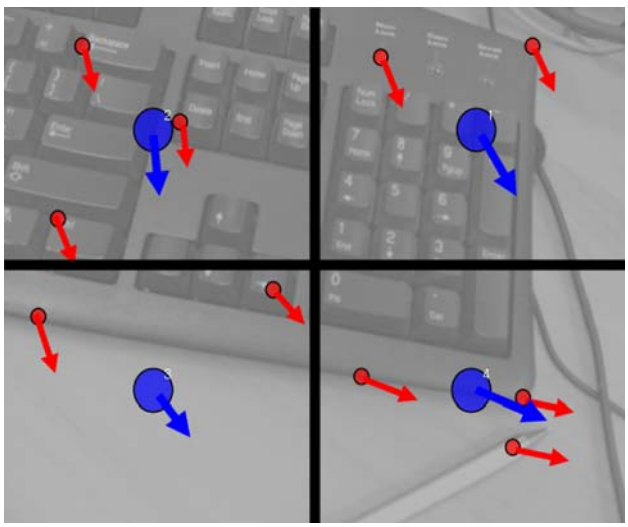

Fig. 11 Finding the rotation factor. The *big arrow* in each quadrant's centre is the mean of the *small elongated arrows*. The algorithm compares means of the quadrant 1 and 3, and 2 and 4 to determine an existence of a rotation

5.2 Tracking performance

The utility and usability of the proposed methods depend highly on the tracking performance. We have measured the accuracy of the motion flow-based tracking, comparing it that of an accurate ultrasonic 3D tracker. Figure 12a shows the motion flow-based tracking performance for translation along the x-axis on the display enacted by horizontal rotation (i.e. rotation around the y-axis, yaw). A 40 inch large display was used with the phone interaction at a nominal (at which the full screen is visible through the camera) distance ($\sim$1.5 m) of from the screen. The camera phone was rotated around the y-axis at seven different angular velocities. For each angular velocity, 200 samples of ideal (from the ultrasonic 3D tracker) and actual (from our method) cursor movement data were collected. After the collection, the ideal cursor position was compared with the actual position by using the average pixel difference.

As depicted in the figure, the tracking error stayed below around 3 pixels up to the motion speed of about 30°/s, and increases sharply beyond this point. The average velocity of the hand rotation was measured at about 60°/s, at which the average pixel error was found to be about 20 pixels. This figure translates to about 1 cm for 1,280 × 1,024 resolution display, and from a distance of 1.5 m, provides reasonable performance. Note that based on this performance, objects (e.g. icons, menu items) to be manipulated can be sized appropriately as well. A similar measurement was taken for z-axis rotation and a similar trend is obtained, but with somewhat lower amounts of error, as shown in Fig. 12b. This is due to the rarer occurrences of feature disappearance during z-axis rotation than during horizontal

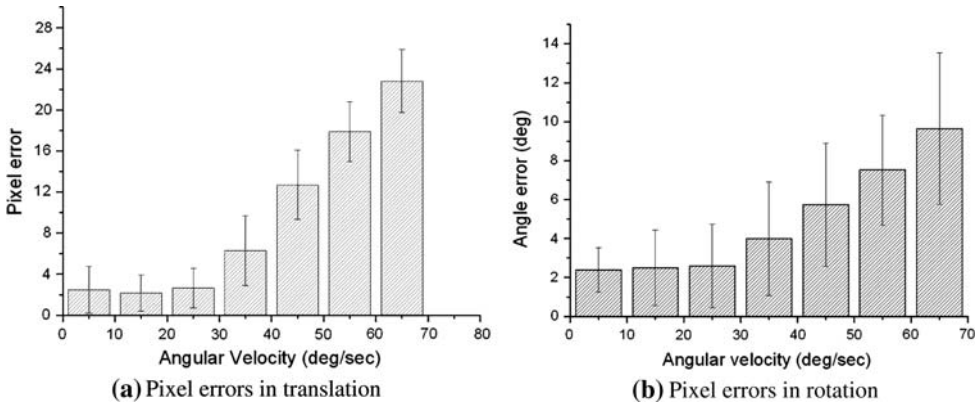

Fig. 12 Pixel errors in horizontal translation (x-axis) and z-axis rotation (roll) at different motion speeds

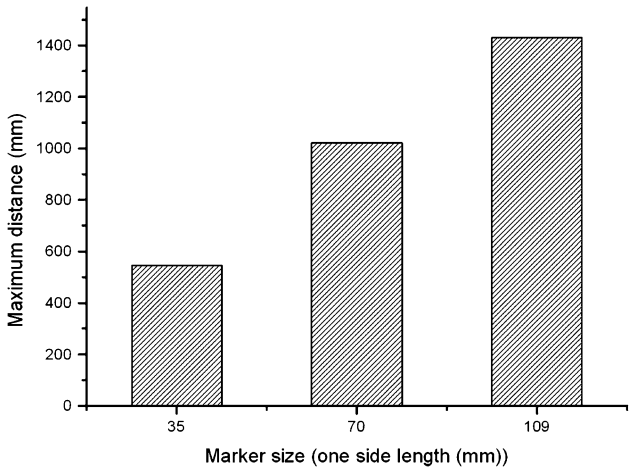

Fig. 13 Maximum distances at which a marker of a given size is reliably recognized

rotation (also see Sect. 6). The large standard deviation, however, indicate lower tracking reliability for horizontal rotation. The average z-axis rotation velocity was measured at about 40°/s at which the angular error was found to be about 8 degrees, also a much more prohibitive performance figure. For instance, rotating a 5 cm object would cause about 12.5 cm rotational error.

The tracking performance of the ARToolkit for marker-based tracking was also measured. The tracking accuracy and marker recognition performance of the ARToolkit marker tracker is affected primarily by relative size of the marker on the camera screen (i.e. once the marker is recognized the tracking performance is less affected). Abawi et al. conducted an extensive experiment of the tracking accuracy of the ARToolkit [33]. According to their work, ARToolkit exhibited about 5–7 cm of tracking error from the distance of 70 cm with a 55 mm marker. This is not sufficient yet for supporting effective manipulation of relatively small objects (e.g. <10 cm objects from 1.5 m distance). Our experience showed that for objects twice as large (or more) than the marker, a reasonable level of interaction efficiency was informally observed.

The relative size is determined by the distance between the marker and the phone, and the size of marker under a fixed camera resolution (160 × 120). Fig. 13 shows the maximum allowable distance at which a marker of three sizes is reliable detected. The three different sizes of marker were set at 64 × 64 pixels (∼35 × 35 mm), 128 × 128 pixels (70 × 70 mm) and 200 × 200 pixels (109 × 109 mm) displayed on a 40 inch display screen (having 1.83 pixels/mm), respectively. The maximum distances reported in the graph may vary with different lighting condition and type of the display device, but the figures still can serve as a guideline for determining the appropriate software marker size given a display size and interaction range. Also note that the marker size should also be set in consideration to the objects they control or manipulate so that the marker does not seriously occlude the objects.

6 Interaction performance and usability

As can be seen from the general results, the current implementation is still far short of supporting recognition of "fast" movements. The camera itself only produces about 10–15 frames of data per second, and the motion flow algorithm about 6–7 frames of tracked data per second. Thus the gesture recognition or tracking performance depends highly on the speed of the gesture/motion. For example, if the mobile phone is rotated faster than 60°/s, one-third of the area of the frame (and corresponding feature candidates) disappears from last frame (the field of view of the camera is about 35°). A related problem is the accuracy. Fine continuous motion control (e.g. at the level of few millimeters or degrees) is not currently possible.

While the high level interaction model should normally be independent of any underlying implementation, in this case, the limitations in each tracking technique affects interaction performance. The tracking accuracy is somewhat improved with the use of markers, however, the

marker-object and marker-cursor-based approaches can cause occlusion of the target interaction object. This problem can be alleviated in part by showing the marker only when needed for a very short time interval (as is in marker-object based approach) and by reducing the size of the marker as small as possible (in marker-cursor based approach).

Another problem is that detecting z-axis rotation (yaw) becomes problematic because users tend to rotate the phone around their wrists (instead of around the center of the phone). This produces discrepancies in the intended motion (e.g. rotate) and resulting interaction. Rotating around the wrist can cause feature point disappearance as well. However, it is only a matter of time that such hardware (and system software performance) constraints will disappear. Once the system resources become sufficient to support relatively fast gestures also, the proposed interaction should prove to be truly effective.

The scenario and task analysis have shown that for sharing distant display, the most spatial interactions only require degrees of freedom of less than two (e.g. translation (2D), rotation (1D), scaling (1D)). The objects shared on display are preferably all visible without mutual occlusions. Thus, 3D interactions in full 6 degrees of freedom are not normally necessary. When using the large display in a more intimate way, at closer distances and with less people to share, 3D interactions may be necessary. Our proposal is to use the mobile phone in combination with other means such as the touch screen, reducing the demands on the mobile phone, again, to provide many degrees of tracking freedom. Still, as already mentioned, the tracking accuracy and reliability must be improved to widen the types of possible tasks and user satisfaction.

In a related work, authors of this article have discovered that despite limited amounts of the feedback (e.g. small display size, limited sound, low resolution graphics, etc.), hand-held devices can still exhibit reasonable interaction efficiency through careful interaction design, e.g. by adopting motion or proprioception-based interfaces for physical/spatial tasks and button based for discrete/logical tasks [34]. The target application platform considered in this work is different, i.e. here, the handheld device is used mainly for interaction with a separate large display, while, in the study cited above, the device contains the display itself. However, we believe that the same principle applies in the interaction situations considered here, and previous researches have demonstrated so through formal usability studies [30, 31], which need not be repeated. Our focus is in finding novel interaction methods using a ubiquitous interaction device such as a mobile phone with a less than perfect tracking capability, and whether the principles can hold in the presence of such a limitation.

7 Conclusion

In this article, we have proposed and implemented various ways to interact with a large shared display system using a camera-equipped camera phone in a ubiquitous computing environment setting. Our motivation starts with the fact that mobile phones with cameras have now become a very common platform in our lives and can act as an ideal medium for various interactions in the ubiquitous computing environment. Three main interaction styles (simple mouse-like continuous tracking in 3D space, embodied agent, and gesture driven) were proposed with two implementations of camera based tracking (e.g. movement flow and marker recognition). Such devices and interaction can further be enhanced with multimodal displays and additional sensing. This work is only the first step to our research in using a multi-purpose handheld device such as a mobile phone as an interaction device. In the future, we plan to evaluate these techniques using rigorous user studies, comparing them to other possible large screen interaction methods. We will also explore other alternative computer vision-based input methods using mobile phones.

Acknowledgments The Lucas–Kanade feature tracker for Symbian OS was kindly provided by ACID (Australian CRC for Interaction Design) for our implementation results. This research is financially supported by the Ministry of Knowledge Economy (MKE) and Korea Institute for Advancement in Technology (KIAT) through the Workforce Development Program in Strategic Technology.

References

1. Norman DA (1998) The invisible computer. MIT Press, Cambridge
2. Wisneski C, Ishii H, Dahley A, Gorbet M, Brave S, Ullmer B, Yarin P (1998) Ambient displays: turning architectural space into an interface between people and digital information. In: Proceedings of the first international workshop on cooperative buildings (CoBuild '98), pp 22–32
3. Lashina T (2004) Intelligent bathroom. In: European Symposium on Ambient Intelligence, Eindhoven, Netherlands
4. Fitzmaurice GW, Zhai S, Chignell MH (1993) Virtual reality for palmtop computers. ACM Trans Info Syst 11(3):197–218
5. Watsen K, Darken RP, Capps M (1999) A handheld computer as an interaction device to a virtual environment. In: Proceedings of the third immersive projection technology workshop
6. Kukimoto N, Furusho Y, Nonaka J, Koyamada K, Kanazawa M (2003) Pda-based visualization control and annotation interface for virtual environment. In: Proceeding of 3rd IASTED international conference visualization, image and image processing
7. Mantyla V-M, Mantyjarvi J, Seppanen T, Tuulari E (2000) Hand gesture recognition of a mobile device user. In: Proceedings of the IEEE international conference on multi-media and expo, pp 281–284
8. Bayon V, Griffiths G (2003) Co-located interaction in virtual environments via de-coupled interfaces. In: Proceedings of HCI international, pp 1391–1395
9. Hachet M, Pouderoux J, Guitton P (2005) A camera-based interface for interaction with mobile handheld computers. In:

Proceedings of the symposium on interactive 3D graphics and games, pp 65-72
10. Lourakis M, Argyros A (2005) Efficient, causal camera tracking in unprepared environments. Comput Vis Image Underst 99(2):259–290
11. Wagner D, Schmalstieg D (2003) First steps towards handheld augmented reality. In: Proceedings of the 7th international conference on wearable computers, p 127
12. Paelke V, Reimann C, Stichling D (2004) Foot-based mobile interaction with games. In: ACM SIGCHI international conference on advances in computer entertainment technology (ACE), pp 321–324
13. Hachet M, Kitamura Y (2005) 3D interaction with and from handheld computers. In: Proceedings of the IEEE VR 2005 workshop: new directions in 3D user interfaces, pp 11–14
14. Hansen TR, Eriksson E, Lykke-Olesen A (2005) Mixed interaction space: Designing for camera based interaction with mobile devices. In: Proceedings of ACM CHI 2005 conference on human factors in computing systems, pp 1933–1936
15. Kruppa M, Krüger A (2003) Concepts for a combined use of personal digital assistants and large remote displays. In: Proceedings of simulation und visualisierung, SCS Publishing House e.V, San Diego, pp 349–362
16. Ballagas R, Rohs M, Sheridan JG (2005) Sweep and point & shoot: phonecam-based interactions for large public displays. In: Conference on human factors in computing systems, pp 1200–1203
17. Rohs M, Zweifel P (2005) A conceptual framework for camera phone-based interaction techniques (PERVASIVE 2005). Lect Notes Comp Sci 3468:171–189
18. Madhavapeddy A, Scott D, Sharp R, Upton E, The Spotcode project website. http://www.cl.cam.ac.uk/Research/SRG/netos/uid/spotcode.html
19. Miyahara K, Inoue H, Tsunesada Y, Sugimoto M (2005) Intuitive manipulation techniques for projected displays of mobile devices. In: Conference on human factors in computing systems, pp 1657–1660
20. Wu M, Balakrishnan R (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proceedings of the ACM UIST, pp 193–202
21. Shen C, Vernier F, Forlines C, Ringel M (2004) DiamondSpin: an extensible toolkit for around the table interaction. In: Conference on human factors in computing systems, pp. 167–174
22. Cavens D, Vogt F, Fels S, Meitner M (2002) Interacting with the big screen: pointers to ponder. In: Conference on human factors in computing systems, pp 678–679
23. Regenbrecht H, Haller M, Hauber J, Billinghurst M (2006) Carpeno: interfacing remote collaborative virtual environments with table-top interaction. Virtual Real Syst Dev Appl 10(2):95–107
24. Maringelli F, Mccarthy J, Slater M, Steed A (1998) The influence of body movement on subjective presence in virtual environments. Hum Factors 40(3):469–477
25. Nokia 6630 symbian OS phone. Avaiable at http://www.symbian.com/phones/nokia_6630.html
26. Kato H, Billinghurst M (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd international workshop on augmented reality, pp 85–94
27. NextWindow 2100 series touch frame. Available at http://www.nextwindow.com/products/2100
28. Bouguet JY (2003) Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm Intel Corporation, Intel Corporation, Microprocessor Research Labs
29. Shi J, Tomasi C (1994) Good features to track. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 593–600
30. Chen M, Mountford SJ, Sellen A (1988) A study in interactive 3D rotation using 2D control devices. In: Proceedings of SIGGRAPH'88, pp 121–129
31. Jacob I, Oliver J (1995) Evaluation of techniques for specifying 3D rotations with 2D input device. In: Proceedings of human computer interaction, pp 63–76
32. Nintendo Wii. Available at http://wii.nintendo.com
33. Abawi D, Bienwald J, Dorner R (2004) Accuracy in optical tracking with fiducial markers: An accuracy function for ARToolKit. In: Proceedings of IEEE and ACM international symposium on mixed and augmented reality, pp 260–261
34. Yim S, Hwang J, Choi S, Kim GJ (2007) Image browsing in mobile device using user motion tracking. In: Proceedings of the international symposium on ubiquitous virtual reality