**Camera-based Interactions for Augmented Reality**

Tatu Harviainen  
VTT Technical Research Centre of Finland  
P.O. Box 1000  
FI-02044 VTT  
+358 400 254078  
tatu.harviainen@vtt.fi

Otto Korkalo  
VTT Technical Research Centre of Finland  
P.O. Box 1000  
FI-02044 VTT  
+358 40 5319674  
otto.korkalo@vtt.fi

Charles Woodward  
VTT Technical Research Centre of Finland  
P.O. Box 1000  
FI-02044 VTT  
+358 40 5001514  
charles.woodward@vtt.fi

**ABSTRACT**

We investigate camera-based interaction techniques suitable for generating simple, easy-to-use augmented reality applications. All the interaction techniques described are based on using only the camera as input device, so that users can interact with 3D content of the application by gesturing with camera movements. With two test applications, we have identified several camera-based interaction techniques that work well with interactive augmented reality 3D content and are intuitive to use and reliable to detect. The test applications are interesting also in their own right, demonstrating technical features such as markerless tracking as well as mobile phone implementation.

**Categories and Subject Descriptors**


**General Terms**

Design, Experimentation, Human Factors.

**Keywords**

augmented reality, interaction, gestures, animation, tracking.

**1. INTRODUCTION**

In this paper we investigate how user interaction with Augmented Reality (AR) could be accomplished using just the camera as interaction device. In the basic use scenario, the user views a tabletop AR application using a hand held device (UMPC, PDA or camera phone), or web camera attached to tabletop PC.

In all the cases, interaction with the application does not require any other physical input devices (e.g. keyboard or mouse) than the camera, nor any user interaction components (e.g. menus, icons, or multimodal ones). As result, the camera-based interaction techniques make the applications extremely straightforward to launch and to use, as there are no buttons or commands to memorize, or to reach out for while operating the application.

To demonstrate the interaction techniques we have chosen two very different kinds of applications: 1) an entertainment application to play with an animated 3D character, and 2) an architectural application linking 3D models to 2D floor plans, intended for professional use. Using these as examples we explain several camera-based interaction cases, which are further divided into several generic interaction classes.

The remainder of the article is organized in further five sections. In Section 2, we take a look at previous work conducted in this area. In Section 3, an overview is provided of our test applications, which is then extended in Section 4 into a more detailed description of the camera-based user interaction methods. In Section 5 we describe the implementation of our test applications using markerless tracking and camera phones. Conclusions and directions for future work are presented in Section 6.

**2. RELATED WORK**

Our work relates generally to 3D user interfaces and gestural interaction, cf. Bowman et al. [1], and in particular to tangible user interfaces, originating from the work of Ishii and Ullmer [2]. The distinguishing feature in our implementation is that the camera/display device itself is the only interaction device involved.

In the field of Augmented Reality, among the first systems presenting tangible user interfaces for object manipulation was the MagicBook by Billinghurst et al. [3], and the “marker paddle” technique by Kato et al. [4]. The work on interacting with the virtual objects with physical markers has later been continued with the “magic cups” technique by Kato et al. [5].

Multimodal extensions to marker-based AR interaction have been studied by Irawati et al. [6], while a study of AR object manipulation using camera phones has been presented by Henrysson et al. [7].

Experiments on using camera phones as see through lenses that augment additional information on physical maps have been introduced for example in Morrison et al. [8] and Rohs and Oulasvirta [9].

Previous work most closely related to ours has been presented by Rohs and Zweifel [10], providing a conceptual framework for
camera-based interactions. Differing from their focus on interaction with physical environment, our work focuses on finding camera-based interaction techniques that work together with real-time augmented 3D content without a need for additional audio or iconic cues on the user interface. Categorization of interaction methods presented in this paper follows the categorization suggested in the forementioned publication.

3. TEST APPLICATIONS

In this section we give a brief overview of two test applications, both featuring multiple camera-based interaction techniques, fine tuned for the context and functionality of the specific application.

The first test application was developed as a marketing tool for a 3D animated TV series “Dibidogs”, produced by Futurecode (Finland) and Blue Arc (China). The content of the TV series is based largely on children's own creativity, which makes it quite unique. The goal for the application was to create an illusion of a 3D animation character, augmented on a marker, being aware of the user and to allow the user to interact with the 3D character in some meaningful way. The final goal for the application is one day to have it implemented on a story book associated to the TV series, each page having different animated content.

In the application, the animated 3D character would react to camera movements and the user could also have the character perform several tricks with gestures that are defined with camera movement patterns. Besides the original marker-based application, we later implemented it also using feature-based tracking directly on the book’s cover page, and finally on mobile phone. See images of the application and animated 3D character in Figure 1.

![Figure 1. Test application with animated 3D character.](image)

The second test application demonstrated AR visualization in bridging the gap between 2D technical drawings and 3D models. The application augmented a 3D model of VTT’s head offices “Digitalo” (i.e. Digital house) over a printed floor plan of the same building, as seen in Figure 2. Earlier AR applications with the Digitalo building have been described in [11].

![Figure 2. (a) 3D model augmented over 2D floor plan (b-c) interaction with cutting planes (d) virtual navigation mode.](image)

4. INTERACTION TECHNIQUES

For the two test applications described in the previous section, several camera-based interaction techniques were implemented. Similar to the [10], the interaction methods are based on physical gestures defined with camera postures. Single postures form a set of interaction primitives used as such to control interactions as described in section 4.1. More complex interaction methods are implemented by combining such interaction primitives. Several complex interaction methods are described in sections 4.2 to 4.4.

4.1 Camera-based Interaction Primitives

The normal use of the camera in AR is to offer a viewpoint for rendering the augmented 3D scene. In camera-based interaction, camera postures can also be used for triggering various interaction events.

In the first test application, camera location and orientation relative to the augmented content were used for driving many of the actions of the 3D character. As the camera was moved, the character’s head turned along to keep facing the camera, and it also tilted in sync with the camera. When tilting too much, the character would roll over and bark. Camera distance from the character also triggered events, such as animation of scratching the ground and sounds of growling when the camera moved to a close distance. Thus, camera transformation based events created an illusion of 3D character being aware of the camera and by doing so, encouraged users to try out different camera movements in a playful manner. This in turn helped users to
discover camera movement pattern gestures, which were also used in the test application.

In the case of the second test application, users could switch between different navigation modes with a camera posture based gesture. When the user rotated the camera so that the horizontal line of the camera image tilted over a predefined threshold angle, application switched from normal AR navigation to virtual navigation mode (see below) and visa versa.

4.2 Camera Movement Patterns
In the first test application, users were able to initiate 3D character actions with camera movement patterns (gestures). For example, vertical shaking motions of the camera triggered an animation in which the 3D character jumped a loop while barking sound was played. Horizontal shake of the camera caused the 3D character to run around in circles while barking.

It is actually quite easy to invent various camera movement patterns, but finding a natural connection between gesture and reaction can be difficult. If the movement pattern has no obvious connection with the triggered reaction, users can not be expected to know how to operate with them without comprehensive instructions or additional UI elements, e.g. icons, audio cues etc. In our case, reliable and error prone gesture detection also proved to be challenging to implement. Gesture detection should be able to adapt to different movement styles of the user and avoid mixing up normal camera navigation with gestural movements. Optical tracking also set limits to the movement pattern recognition, for example fast movements of the camera can easily be lost due to the motion blur in recorded camera image. By accounting for the technical limitations of the optical tracking and by finding natural connection between gesture and reaction, however, camera movement pattern gestures can effectively extend camera-based interaction.

4.3 Controlling 3D Widgets
Camera controlled 3D widgets were used in the second test application in the form of cutting planes. In their initial position, five cutting planes surrounded the 3D model, one plane on each side of the model, except at the bottom. Users were able to select and move cutting planes by moving camera towards the 3D model through a desired cutting plane. As the camera moved through a cutting plane, the plane was activated and the previously used cutting plane was reset. The user could push the activated clipping plane along the normal axis of the plane to a desired position and then move camera back to view the result. Cutting planes proved to be quite intuitive to use and users felt comfortable using them.

4.4 Virtual Navigation Mode
In the second test application, users were able to change the navigation mode from normal augmentation to virtual navigation mode with a simple gesture of rotating the camera. Rotation of the camera so that the horizontal line tilts, is not required in normal operation of the test application, but as a gesture it is easy to perform and to remember.

When mode change is initiated, virtual camera is moved inside the 3D model to the spot pointed earlier by the camera. In this mode, the user can freely move the physical camera to look around the 3D model as if the model was surrounding the camera in full scale.

In virtual reality domain, combining different navigation techniques has been seen as somewhat challenging. According to our experience with the virtual navigation mode, it seems that by this interaction technique an additional navigation mode can be added with relatively satisfying results.

5. IMPLEMENTATION ISSUES
In this section we discuss implementation of our applications on camera phones, as well as some details of our camera tracking solutions.

5.1 Tracking
We used the ALVAR subroutine library [9] to implement both the marker based as well as markerless tracking methods used in the test applications. In the markerless version of the first test application, the user sets the 3D character to the scene by aiming the camera to some flat surface (e.g. comic book cover) and pressing a key. Then, strong corners from the video image are detected, and their positions in world frame are calculated assuming that the camera optical axis is perpendicular to that surface. As new frames are acquired, the corners are tracked in image domain using Lucas & Kanade tracking algorithm, and camera pose is updated by minimizing the reprojection error of the world points. Robust M-estimator is used in optimization procedure to prevent the effect of outliers. New features are detected and their 3D positions in world frame are reconstructed as the camera is moved. The system detects outliers and removes them from the feature list, and uses only the features whose reprojection error is under some fixed limit.

The second test application was implemented using hybrid tracking, where a set of markers are used in conjunction with natural features to determine the camera pose. In this case, the corners of the markers are considered as ideal measurements and they are given strong weights on camera pose optimization procedure. Natural features, which are detected inside the bounding box of the marker set (see figure 2), are tracked in image domain as above. New features are added and reconstructed by assuming that they lie on the same plane as the markers. This allows the user to move the camera freely over the floor plan. If the markers are not visible, the system relies totally on feature-based tracking as described in first case. As one or more markers become visible again, the drift is compensated and the outlier features are deleted.

Tracking used in virtual navigation mode is implemented by calculating the optical flow from the video, and assuming that the camera translation is zero and the rotation is done in yaw and pitch directions. Under these assumptions, the mapping between the optical flow and camera rotation is straightforward to calculate as camera field-of-view is known.

5.2 Camera Phones
To try out our interaction methods on different kinds of display devices, the first test application was ported to Symbian operating system. The implementation was done for Symbian
S60 3rd edition phones and tested on a Nokia N95 8G phone, seen on figure 3.

Due to the limited processing power and different 3D graphics library of the mobile platform, only marker-based tracking and limited animation features were implemented on the phone version of the test application. By adjusting 3D model and animation resolutions suitably, our test application provided frame rates of approximately 7 fps on the phone device.

However, feature limitations did not change the basic use scenario nor interaction methods of the application. With the phone version, we were able to confirm the usability of the described interaction techniques also on a moderate performance mobile device.

Figure 3. First test application on mobile phone.

6. CONCLUSIONS

In this paper several camera-based interaction techniques for AR have been introduced. Camera postures relative to the augmented content are used as such to control interactions and/or combined to achieve several additional interaction methods:

1) Camera movement patterns; used in the first test application for triggering different behaviors of the character.

2) Controlling of 3D widgets; used in the second test application for controlling the clipping planes.

3) Mode changes; used in the second test application for switching to virtual navigation mode within the model.

Although the success of the proposed techniques has not been validated with formal usability studies of any kind, based on our use experiences, they appear to be easily accepted by users and applicable for many kinds of simple AR applications. With the introduced test applications users were able to operate successfully in minutes without explicit instructions concerning the UI or application features.

Based on our experiments with different display devices, it appears that the proposed interaction techniques offer quite similar use experience on mobile phones as on handheld PC, or PC with external camera. However, HMD usage differs quite radically, and the proposed interaction techniques are not ideally suited for use with HMDs.

Generality of the described interaction techniques across other applications has not been directly addressed in our research, but we hope our discussion gives useful ideas also for others developing easy to use and natural interaction methods for AR applications. In the near future, we look forward to extending our research with the goal of developing more comprehensive camera-based interaction categorization and in engaging formal usability studies on proposed interaction techniques.

7. REFERENCES


