Foot-based mobile Interaction with Games

Volker Paelke, Christian Reimann, Dirk Stichling
University of Paderborn, C-LAB
Fürstenallee 11
33102 Paderborn, Germany
E-mail: {vox; reimann; tichel}@c-lab.de

ABSTRACT

Interaction with mobile applications is often awkward due to the limited and miniaturized input modalities available. Our approach exploits the video capabilities of camera equipped smart-phones and PDA's to provide a fun solution for interaction tasks in simple games like “Pong”, “Break-out” or soccer.

THE CONCEPT

Many simple games rely on a basic position input from the user as their central control paradigm. For example, in the classic game “Pong” two users try to position their paddles so that the ball is kept from leaving the playing area. Similar control paradigms are used in break-out games like the classic “Arkanoid”.

In foot-based mobile interaction we propose to use the camera of current video capable mobile devices to detect motion and position of the user’s foot to effect the input that is required for such games.

The camera facing towards the back of the mobile device (Fig. 1) is used to detect "kicking" movements of the user's feet. When a collision between the user's "kick" and an interaction object shown on the screen of the mobile device is detected, a corresponding interaction event for the application is generated.

Figure 1: PDAs with mounted/integrated camera

In games like “Pong” and “Arkanoid” the interaction object is the ball that is moving across the playing area. Depending on the specifics of the game under consideration either only the position of the foot is used (e.g. in “Pong”) or the direction and speed of the detected motion can also be exploited.

The challenge is to analyze the video stream from the camera in real-time. While this offers a completely new possibility for interaction, processing time and lag must be minimized for effective interaction with corresponding requirements on the processing power. A common approach is the use of a client-server architecture that we have examined in the AR-PDA project (see ARPDA2003). In this approach the client (the mobile device) only captures the camera-image, sends it to the server where the costly image processing and analysis operations are performed. The resulting data is then transmitted back to the client.

For mobile gaming this is obviously not a suitable option, because of several important drawbacks. First of all a working broadband-network connection is required at all time, which can pose economical problems due to connection costs as well as technical problems. The biggest technical problem is the latency caused by the network transmission and the incorporated encoding and decoding. For technical illustrations (such as used within the AR-PDA project) the latency (in some cases more than 1 second) is already a problem, but even more so for games, which are often highly interactive and need to be very responsive.

We have therefore developed a fast computer vision algorithm that is capable of running in real-time on current mobile devices like PDAs and detects foot position and motion from the video input.

IMPLEMENTATION

Our computer vision (CV) component calculates the motion of a "kicking object" (e.g. the user's foot) from the incoming video-stream and detects collisions between it and the interaction objects.

The results of the collision detection are used to calculate the new direction and speed of the interaction object, e.g. the ball. Since current mobile devices lack high-performance processors we have developed a simple and fast algorithm that combines 2D edge extraction and tracking and operates only in regions of interest (ROI) around the interaction objects (see figure 2, left). Due to performance issues instead of calculating the optical flow of the ROI we have developed a fast 2D edge extraction and tracking algorithm. This algorithm is presented in (Stichling and Kleinjohann, 2003).

It is based on a design methodology developed for real-time CV algorithms called CV-SDF (Computer Vision Synchronous Data Flow) (Stichling and Kleinjohann, 2002).

To perform the collision detection, straight edges inside the ROI are vectorized and tracked between two consecutive images. The median direction and speed of the motions of all edges inside the ROI is computed afterwards. If the median direction points towards the interaction object an interaction event is generated that is then used in the game, e.g. to update the speed and direction of the virtual ball.

Figure 2: Edge-Detection and Motion-Tracking

To perform the collision detection, straight edges inside the ROI are vectorized and tracked between two consecutive images (Fig. 2, right). Then the median direction and speed of edge motions inside the ROI is computed. If it points towards an interaction object an interaction event is transmitted to the application

EXAMPLE

To demonstrate the function of our algorithm and its application in games we have developed a small game called AR-Soccer (Fig.3). In this game the player has to kick a virtual ball (a moving interaction object) with his real foot into a virtual goal, bypassing the goalkeeper.

Figure 3: AR-Soccer and the edge-tracking algorithm

The AR-Soccer game can be viewed as a single-player variant of "Pong" and the same interaction mechanism can be used in break-out games like "Arkanoid".

EVALUATION

The current prototype of our system was informally evaluated by more than 30 users (both experienced in IT and not) with different educational backgrounds. The overall feedback of the users was very positive. The users had no general problems with the CV based interaction and found it very intuitive. The main problem was the reliability of the computer vision, which was caused by the low framerate. Because the program runs at only 5-6 fps, very fast movements can not be recognized. A test with a PC-version of the game (holding the camera in hand and looking at the screen on a table), which was much faster, has shown that the occurring problems are caused by the low framerate. Nevertheless after a short time most users get used to the restriction and kick the ball more slowly. Most test users pointed out that the more physical interaction of standing and kicking (compared to just sitting and pressing buttons) caused a lot of fun and made the game much more exciting. But especially the older users were often too shy to play so active, because they were afraid to look too ridiculous. Nevertheless, they tried pretty hard to win the game when they felt unwatched.

OUTLOOK

While current hardware limits our system to the processing of 5-7 frames/second, suggesting its use as part of games where the interaction is part of the challenge, feedback from users of the AR-Soccer application has been very positive. Up to now we have considered only single player applications. In the future we plan to use the wireless networking capabilities of mobile devices to extend the approach to two-player and multiplayer games, e.g. the classic two-player "Pong".

REFERENCES

1. Stichling, D. and Kleinjohann, B.: Edge Vectorization for Embedded Real-Time Systems using the CV-SDF Model, *Proc. Vision Interface 2003* , Halifax, Canada
2. AR-PDA (2003): The AR-PDA, a personal digital mobile assistant for virtual and augmented reality, Project-homepage http://www.ar-pda.com
3. Dunlop, M. D. and Brewster, S. A. (Eds.) (2001): Proceedings of Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices, IHM-HCI 2001 Lille, France, September 2001
4. Johnson, C. (Ed.) (1998): Proceedings of the First Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1, University of Glasgow, Scotland
5. Nölker, C. and Ritter, H. (1999). GREFIT: Visual Recognition of Hand Postures. In Proc. Int. Gesture Workshop GW '99, 1999 , pp. 61-72.
6. Stichling, D. and Kleinjohann, B. (2002): {CV-SDF} - A model for Real-Time Computer Vision Applications, in proceedings of WACV 2002: IEEE Workshop on Applications of Computer Vision, Orlando, FL, USA, December 2002
7. WMCSA (2002): Workshop on Mobile Computing Systems and Applications, IEEE, http://wmcsa2002.hpl.hp.com/