Video See-Through and Optical Tracking with Consumer Cell Phones

Mathias Möhring, Christian Lessig and Oliver Bimber
Bauhaus University

1 Motivation

To enable mobile devices, such as head-mounted displays and PDAs, to support video see-through augmented reality is a popular research topic. However, such technology is not widely-spread outside the research community today. It has been estimated that by the end of the year 2005 approximately 50% of all cell phones will be equipped with digital cameras. Consequently, using cell phones as platform for video see-through AR has the potential of addressing a brought group of end users and applications. Compared to high-end PDAs and HMDs together with personal computers, the implementation of video see-through AR on the current fun- and smart-phone generations is a challenging task: Ultra-low video-stream resolutions, little graphics and memory capabilities, as well as slow processors set technological limitations. We have realized a prototype solution for video see-through AR on consumer cell phones. It supports optical tracking of passive paper markers and the correct integration of 2D/3D graphics into the live video-stream at interactive rates. We aim at applications, such as interactive tour guiding for museums and tourism, as well as at mobile games.

2 Light-Weight Optical Tracking

The main objective of the implemented image analysis algorithm is to determine four non-coplanar points on a 3D paper marker. They are used as input parameters for rendering the overlaid 3D geometry correctly using a weak perspective projection to model the camera-to-image plane transform (see section 3). Using such a concept instead of determining the extrinsic parameters of the phone's camera relative to the marker ensures that our approach does not depend on the camera's intrinsic parameters. Thus it can be applied in combination with any camera without having to correct lens distortion and measuring focal lengths. In addition, the algorithm has to recognize individual markers by reading a circular barcode. Using colored features, such as lines and circles on the marker (cf. figure 1) allows searching for increases in RGB channels instead of having to detect a specific color or greyscale. This makes the tracking process robust and – in combination with a specific spatial distribution of the features on the marker – allows encoding a large number of different codes. To achieve interactive frame rates, a multi-level scanning algorithm has been developed: Each frame is scanned with continuous, parallel scan lines in every *n*-th pixel-row or column until an expected color increase has been detected in two consecutive scan lines. Two pixels of the same color but from different scan lines are part of the same line feature if the color of the pixels lying between them is identical. This is repeated until the endpoints of the line feature have been reached. One of the two endpoints is the origin point of the marker. The other two line features are traced from the origin to detect the remaining two endpoints. The result of this is the projections of three endpoints and one origin point that are non-coplanar on the 3D marker. These points span rectangular reference frames in the image that are sampled for the projections of circular features on the marker using a simple quad-linear interpolation. Spatial and color appearance of circular features lead to a predefined index code. Furthermore, interframe information is used to narrow the search region and to eliminate impossible results.

3 Affine Object Representation

To render the overlaid 3D graphics in the correct perspective relative to the cell phone and the marker, the geometry has to be transformed into a global affine coordinate frame. This coordinate frame is defined by the four non–coplanar basis points determined during the optical tracking step. The pixel positions of vertices can be approximated by projecting them with respect to the detected pixel positions of the four basis points. This approximation can be fully integrated into the common OpenGL rendering pipeline (like in OpenGL ES). Thereby the first two rows of the projection matrix are filled with vectors, calculated from the *u* and *v* pixel coordinates of the basis points, affecting *x* and *y* coordinates of each vertex and simultaneously spanning the virtual camera's plane in the affine coordinate system. The third row is defined by the cross product of the other two vectors. This vector, however, does not provide correct depth information, but provides the correct depth ratio between all vertices. To use this matrix in the OpenGL transformation pipeline, an additional scaling has to transform the resulting values into normalized device coordinates and the correct depth range. While the camera image is displayed in the background, 3D graphics is overlaid by rendering it into the affine coordinate system using the customized OpenGL pipeline. The lack of a floating point unit on consumer cell phones requires a fast fixed point arithmetic that avoids divisions.

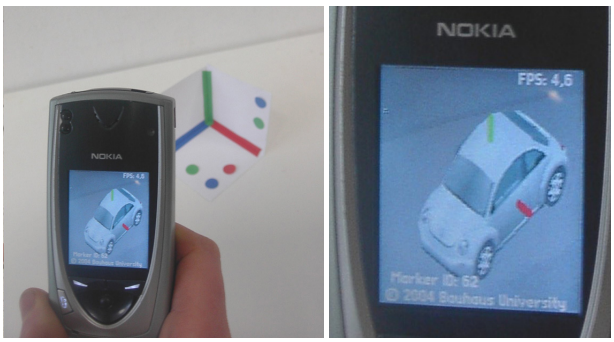

Figure 1: 3D graphics rendered perspectively correct into the live video stream with respect to the position of the cell phone relative to the marker.

References

VALLINO, J. and KUTULAKOS, K.N. 2001. Augmented Reality Using Affine Object Representations. *Fundamentals of Wearable Computers and Augmented Reality*, pp. 157-182, ISBN: 0-8058-2902-4, Lawrence Erlbaum (publisher).

e-mail: {moehring, lessig, bimber}@uni-weimar.de
url: www.uni-weimar.de/medien/AR