IPT-EGVE Symposium (2007)
B. Fröhlich, R. Blach, and R. van Liere (Editors)

A Lightweight ID-Based Extension for Marker Tracking Systems

Daniel Flohr[1†] and Jan Fischer[2 *]

[1]WSI/GRIS, University of Tübingen, Germany [2]Island Graphics Group, University of Victoria, Canada

Abstract

The estimation of the position and orientation of the digital video camera is a central challenge in video see-through augmented reality. Many augmented reality applications solve this problem with the help of marker-based methods, which analyze artificial fiducials in the images of the real environment, e.g., using the widespread ARToolKit library. Among the drawbacks of conventional marker tracking is the necessity to manually define marker patterns. Badly chosen patterns have a negative impact on tracking performance. Although improved methods for automatic marker generation have been described, manually controlled marker tracking is still widely used in many applications for practical reasons. In this paper, we describe a lightweight drop-in extension for ID-based marker tracking. Our system makes it possible to automatically generate a large number of tracking fiducials identified by unique numerical IDs. The created marker patterns consists of large monochrome patches, which improves the recognition rate and tracking performance compared to typical manually defined fiducials. Due to the design of our extension, only minimal adaptations are required in order to add ID-based tracking to existing augmented reality software. We discuss experimental results demonstrating the improved pattern recognition and describe an example application.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Artificial, augmented, and virtual realities

1. Introduction

Over the course of the last decade, augmented reality (AR) has become a growing field of research. AR systems superimpose computer-generated visual information over a view of the real world. AR applications have begun to be used beyond their former almost exclusively scientific existence, and potential practical applications of AR range from soccer television broadcasts to entertainment software [NLL04] to medical applications [FNFB04]. An overview of possible AR applications and the principles of various tracking techniques was published by Azuma [Azu97].

One of the main challenges in augmented reality is the real-time estimation of the position and orientation of the viewer or the digital video camera used in the system. A widespread solution to this problem is marker tracking. Marker tracking systems use a set of physical patterns in order to identify objects and to establish the geometric relationship between virtual and real objects in an AR environment. In video see-through AR applications, one or more cameras provide video frames for image analysis. Each frame is examined to find visible potential markers. The marker candidates are then compared to marker patterns stored in a database. After a marker has been recognized, a relationship between this marker and the camera can be estimated. Image based tracking libraries calculate camera position and camera orientation relative to physical markers in real time. The resulting camera pose estimation allows for a perspectively correct integration of virtual content into the video stream. One widely used implementation of marker tracking, the ARToolKit [KB99], is a software library for building AR applications, allowing for easy development of a wide range of augmented reality applications.

The generation of marker patterns often times involves many steps of user interaction, and the quality of the marker patterns depends on the lighting and the camera hardware during the generation process. By contrast, ID-based marker generation allows for easy generation of large amounts of markers with consistent quality. Here, we present a new system for ID-based marker generation and management, which we have named *binARyID*. *BinARyID* uses bitstrings to represent and generate physical markers and improves usability and stability for marker tracking systems. Although currently implemented as an extension to ARToolKit, *binA-*

† e-mail: daniel@realitynet.de *e-mail: jan@janfischer.com

RyID can easily be adapted to other marker tracking systems. The lightweight and portable single class implementation of our approach combines ID-based tracking features with an easy integration into existing projects. With this extension, some features known from advanced technologies like ARTag [Fia04] or ARToolKit Plus [WS07] are available for the widely used ARToolKit, and some of their features can be added to other marker tracking systems.

Designed as an extension to existing tracking systems, *binARyID* inherits the shortcomings of the underlying marker tracking algorithm. It cannot improve the sensitivity to occlusion or partial visibility of markers found in conventional marker tracking technologies. However, unlike other advanced vision-based tracking technologies, our approach is implemented as a small, self-contained extension to marker tracking systems. It does not require the modification of the ARToolKit source code, and only a single additional class file has to be added to existing application code. Using *binARyID*, legacy AR applications can be modified with minimal effort to use ID-based marker patterns. We view this extreme ease of use, portability, and fast integration into existing projects as significant advantages.

2. Related Work

The ID-based marker generation presented in this paper is meant to be used with augmented reality applications. An excellent overview of AR related topics and AR applications is given by Azuma [Azu97]. Vision-based marker tracking has been described for example by Kato and Billinghurst in [KB99]. The use of ID-based markers in the ARTag system was presented by Fiala [Fia04]. A similar approach is taken by ARToolKit Plus, which has been actively developed for many years [WS07]. A comparative evaluation of several marker tracking systems can be found in [ZFN02]. Moreover, markerless optical tracking methods have been developed, e.g., the approach described by Reitmayr and Drummond that allows for optical tracking without the use of physical markers [RD06].

Non vision-based tracking systems exist and are widely used. Infrared tracking is used for motion capturing in computer animation as well as for AR applications [PP04]. The global positioning system (GPS) uses radio signals and has been employed in AR applications for tracking purposes, e.g., in ARQuake [TCD*00]. Hybrid tracking combines multiple tracking methods and can significantly improve tracking quality and reliability. An example of a hybrid tracking system for surgical applications was presented by Fischer et al. in [FEBS07].

3. Marker Tracking

Marker tracking algorithms evaluate camera images to establish a relationship between the real environment and virtual components in an augmented reality system. Marker tracking systems use a set of predefined marker patterns. These patterns correspond to images printed on physical markers, which are placed within the scene. Each marker establishes its own coordinate system. Any time a marker is visible in a captured video frame, a transformation matrix from the coordinate system of the physical marker to the current camera coordinate system is estimated. This is done by real-time image analysis. Typical marker tracking algorithms use multiple steps for the extraction of a marker from the current video frame. In a first step, only contours are taken into consideration. Although circular shaped markers have been used for tracking [FN03], most systems use quadratic markers. Contours of quadratic markers can be described by four line segments under any given camera angle. The edge vertices of the contour are determined by intersections of two line segments, and their coordinates are stored. Image regions containing matching contours are extracted and compared to marker patterns stored in the pattern database of the tracking system.

In theory, any image can be turned into a marker pattern, but simple shapes like letters or geometric shapes typically provide better tracking quality than complex shapes and images. Due to limited video resolution during the capturing process, simple shapes are identified by the tracking system much easier than more complex shapes. This is especially true for extreme camera angles. Fig. 1 illustrates the advantage of simple shapes in terms of recognition. The triangle shape is easier to recognize under a low camera angle. The more complex shape on the left suffers from aliasing effects, and small details in the pattern can not be distinguished.

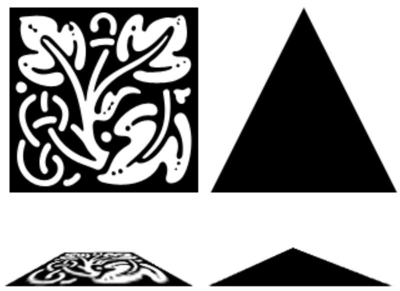

Figure 1: *Complex shape compared to simple shape. Straight on views and low camera angle.*

The typical process for generating marker patterns for AR applications consists of several steps. Every marker has to be defined and created by the user in a pre-processing step, requiring additional tools for image creation and manipulation. Afterwards, the marker is printed and then read into the tracking system by capturing it with a camera.

The standard method for making these captured pattern files available to ARToolKit applications consists of several distinct steps. All pattern files have been loaded, manually assigning a specific pattern size and an ID. The complex and repetitive process of creating the marker images and pattern files becomes a serious issue when working with large amounts of markers. In addition to the tedious marker creation, printing, and the capturing of the pattern with a camera, any exchange or the addition of new markers usually implies manual file management.

The capturing of the internal pattern data-set using a camera bears additional issues. The user has to be very cautious in terms of consistency of lighting and camera position during the marker capturing. Lighting and camera noise normally alter the way color values and brightness are represented in the resulting internal representation. This may lead to unstable tracking in a variety of cases, caused by a change in lighting, camera angle or the exchange of the camera itself.

These issues spawned the idea of creating a lightweight extension to existing marker tracking systems, taking care of almost all the steps of creating and handling marker and pattern files. Although currently implemented as an ARToolKit extension, the concepts of *binARyID* can easily be adapted to other marker tracking systems.

4. ID-Based Marker Generation

The idea behind the ID-based approach presented here is to automatically generate markers based on simple geometric shapes, which can be robustly identified by the marker tracking algorithm. In our approach, each marker consists of 16 quadratic fields arranged as a 4x4 matrix. Every single field of this matrix can be either completely black or completely white. Using monochromatic markers makes the system more resistant to camera noise, color bleeding effects and the use of different camera hardware. The resulting patterns are shapes with enough potential variation to generate a large number of different patterns while still being simple enough to guarantee excellent image based tracking under most conditions.

Each marker can be represented by a 16 bit binary number, which we call the binary ID. Every monochrome field of the marker is associated to one bit of the binary ID. The ID of a marker is used to generate black and white images for marker tracking applications. Similar approaches are also used in advanced marker tracking systems like ARToolKit Plus [WS07] and ARTag [Fia04]. A comparison of a typical manually defined ARToolKit marker and an ID-based marker generated by our system is shown in Figure 2.

The ID-based pattern generation core of *binARyID* can be used in two different ways. Working as a stand-alone application, it generates markers suitable for almost any AR marker tracking environment. In this mode, it generates a user defined number of marker images ready for printing. Used as ARToolKit extension, on-the-fly ID-based marker creation and file handling is now available for ARToolKit applications. This extension is not altering the core of ARToolKit, making it extensible and adaptable to future ARToolkit versions or even other image based marker tracking systems. Existing applications can easily be adapted to benefit from the extended functionality and improved usability.

Figure 2: *Traditional ARToolkit marker (left) compared to ID-based marker (right).*

The generation of ID-based markers is implemented in a multiple step pipeline, which begins with the generation of a 16-digit bitstring from a manually specified or automatically generated integer input ID. Next, a 4x4 pattern matrix is generated from this bitstring. The basic structure of a binary ID matrix is shown in Figure 3. 16 bit marker IDs theoretically allow up to $2^{16} = 65536$ different IDs. However, more than two thirds of the generated IDs have to be discarded during the marker creation process. In interactive AR applications, users manipulate both real and virtual components in addition to moving within the real world environment. Due to the resulting free placement and rotation of markers in the camera image, it becomes clear that the numerical uniqueness of an ID bitstring is not strict enough a criterion for a marker to work in interactive image based tracking scenarios.

0	1	0	1
1	0	0	0
0	0	0	0
0	0	0	0

Figure 3: *Matrix representation of a binary ID.*

To be viable for error-free marker tracking, every pattern has to be unique from any given camera angle. Therefore, every pattern needs to be invariant in terms of rotation. A pattern may not be identical to itself after one or more ninety degree rotations, as shown in the example in Figure 4. Such a pattern would cause severe errors when being used for tracking.

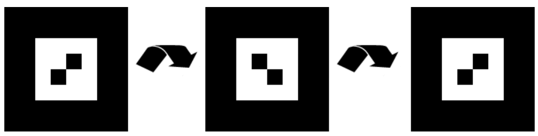

Figure 4: *This marker matches itself after two rotations by ninety degrees.*

Moreover, patterns must not be identical to any previously generated pattern after one or more ninety degree rotations, as illustrated in Figure 5. It would be impossible for the tracking algorithm to decide how the camera is oriented in relationship to such a marker, or to uniquely identify the marker in the first place. Avoiding the generation of non-valid marker patterns is a key aspect for ID-based marker generation.

Figure 5: *This marker is equal to a previously generated marker after one rotation by ninety degrees.*

For this reason, a test for rotation invariance was implemented in *binARyID* in order to reject all patterns not meeting these criteria. Every candidate pattern is rotated by ninety degrees three times by transposing the associated matrix. Every time it is compared to the original matrix and every valid pattern found so far. If no identical pattern is found, the pattern is valid, and the binary ID of this pattern is stored. All valid patterns are also numbered continuously. This way, an integer ID is assigned to every valid pattern. After determining all valid patterns, the image files and ARToolKit pattern files can be generated. Running as an ARToolKit extension class, the system is given an array of integer numbers as input, representing the markers to be used in an AR application. For quick access to all IDs, a header file containing a lookup table for translating the input integer to a binary ID is generated as well. The following pseudo code summarizes the process of finding valid marker patterns:

```
PROGRAM binaryIdPattGen

INPUT Desired number of markers or inte-
ger IDs
OUTPUT Marker images, ARToolKit patterns

FOR every pattern integer DO
    generate binary ID
    generate pattern 4x4 matrix
    FOR 1..3 90 degree rotations DO
        test for rotational symmetry;
        compare with all stored valid markers
        IF (not matching self or previously
            stored markers)
        THEN add pattern to valid markers
    END FOR
 END FOR
 FOR every valid pattern DO
    generate pattern and save to file
    generate and save image file
 END FOR
 END binaryIdPattGen
```

5. Implementation

Our ID-based approach to generating marker patterns improves usability and minimizes tracking errors. Although being currently implemented as an extension for ARToolKit, the principles of *binARyID* are adaptable to other marker tracking systems. The system is capable of creating image files for printing and ARToolKit pattern files for direct use with ARToolKit applications. Moreover, using the drop-in extension, the marker generation can be done from within existing, ARToolKit-based source code. In this case, every marker pattern is generated on the fly during program initialization. The printing of the physical marker images remains the only preprocessing step requiring user interaction apart from placing the markers within the scene or attaching them to physical props. With automated ID-based marker generation, many disadvantages of common marker generation procedures can be avoided. All valid markers are identifiable by a binary (bitstring) ID and an integer ID. As mentioned in Section 4, all valid markers are numbered continuously to generate the integer IDs. Therefore, the integer representations do not directly correlate to the corresponding binary 4x4 matrices. The continuous numbers are additionally provided in order to maximize usability and to prevent users from picking IDs that would have to be rejected during marker generation. The original ARToolKit source code did not have to be changed, which makes the extension easy to use and adaptable to future versions of ARToolKit or other marker tracking software.

6. Results

Some examples of ID-based marker images generated with *binARyID* are shown in Figure 6. ID-based markers behave just like traditional ARToolKit markers and can easily be integrated into existing AR environments. ID-based markers generated with our approach can lead to better and more stable tracking in some conditions, especially under bad lighting conditions and extreme camera angles. We have run experiments under consistent lighting conditions and camera angles. In our experiments, we found differences between typical, manually defined markers used in ARToolKit applications and markers generated with *binARyID*. The single marker experiments described here compared a randomly selected ID-based marker with the standard ARToolKit marker shown on the left of Figure 2. For the multi marker scenario, additional standard markers were created manually.

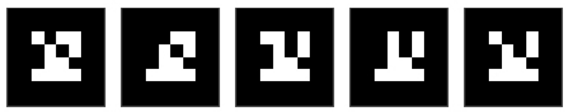

Figure 6: *Some ID-based markers generated by binARyID.*

The first experimental setup was a bad lighting situation, which usually leads to very noisy camera images and a reduced marker detection accuracy. The ID-based markers have proven to work slightly better under these circumstances. The camera was placed at angles from 10 to 80 degrees in relationship to the marker, and the average reprojection error of the marker corners was measured. The reprojection error is defined as the Euclidean distance between the actual marker corner positions in the camera image and the marker corner positions projected with the currently estimated pose. The results of these measurements are shown in Figure 7. The ID-based markers have proven to have a slight advantage over the standard markers, in particular when using extreme camera angles.

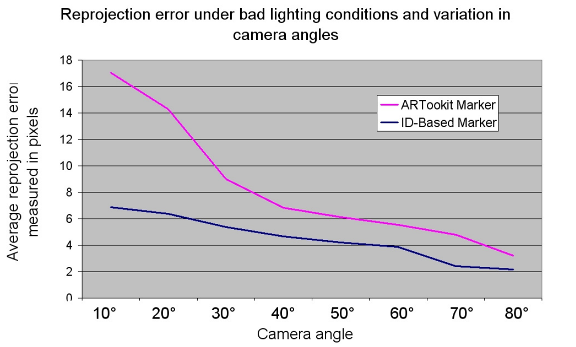

Figure 7: *Measured reprojection error using ID-based markers and standard markers under bad lighting conditions.*

The superiority of ID-based markers was confirmed by a second experiment run under good lighting conditions, again using extreme camera angles. The angle was chosen so that ARToolKit would not be able to recognize the standard marker in every frame. After capturing one hundred continuous frames, the number of failed pose estimations was counted. In this scenario, the ID-based marker was detected in 89% of all frames, whereas the manually defined ARToolKit marker was detected in 71% of the frames.

A third experiment compared both marker types in a multi marker setup. All markers were visible in every frame. The test was run in setups from one to seventeen markers. Under these circumstances, the camera is usually much farther away from the markers than in single marker setups. To be able to frame all markers at the same time, the camera was set up farther away from the scene every time a new marker was added. Figure 8 shows the results of the multi marker experiments. Here, the average number of markers which were not recognized is plotted against the total number of markers in the setup. The results demonstrate hat the use of ID-based markers can lead to improved marker recognition in such applications.

7. Example Application: Tangible Interaction

We have built an example AR application using our ID-based marker system *binARyID*. As a multi marker scenario, a basic urban planning application was implemented, allowing

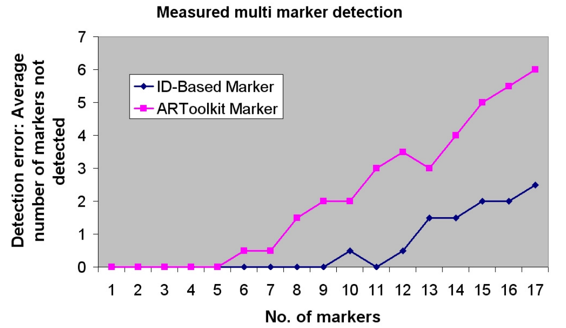

Figure 8: *ID-based markers and standard markers in multi marker setup.*

for tangible interaction with several physical props identified by individual markers. Multiple markers are required to tag props and to compensate for occlusion by other props or user interaction. The application computes shadows cast by buildings and visualizes a simple wind flow simulation in an urban environment. This application is similar to an earlier, non-marker tracking based system designed by Ishii et al. [IBJU*01]. The left of Figure 9 shows the marker setup without any augmentations. The same setup with augmentations is shown on the right of Figure 9.

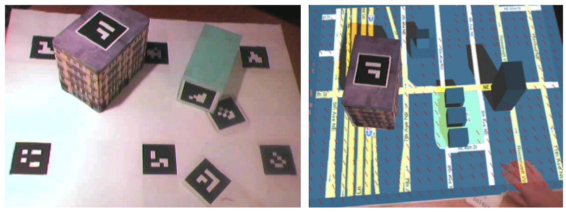

Figure 9: *Example urban planning application requiring several ID-based markers.*

8. Discussion

As shown in Section 6, ID-based markers generated with *binARyID* can improve tracking performance. There are several reasons for this. The automated generation of the markers does not involve an image capturing step. Therefore, the marker images are not influenced by varying lighting conditions and are free from camera image artifacts that standard markers may suffer from. Furthermore, the large monochrome 4x4 matrix patterns are easy to recognize by existing tracking algorithms, while allowing for enough variation to create a large number of different markers with consistent quality. Our markers are strictly monochrome, making them resistant to variations in lighting, color bleeding effects and camera noise. The example application uses multiple ID-based markers in a tangible interaction environment, showing very good tracking performance under different lighting conditions and camera angles.

9. Conclusion

The ID-based marker generation method presented here allows for easy creation of large amounts of markers for augmented reality applications. Although implemented as an ARToolKit extension, the principle is applicable to other AR libraries using marker tracking. Using *binARyID* can significantly improve usability of AR applications. Marker generation and management is completely automated. Tracking quality may increase under certain conditions. The experimental data and the example application as well as our experiences during the implementation have shown the advantages of *binARyID*.

The use of ID-based markers generated with binARyID inherits the shortcomings of the underlying marker tracking algorithm. In the case of the ARToolKit, this is mainly the sensitivity to occlusion and extreme angles or partial visibility due to camera image edges. This, however, is an acceptable trade-off with the advantages of a lightweight and portable extension.

The lightweight drop-in extension for ID-based marker tracking described here inherently cannot deliver the same robustness as more advanced tracking technologies like ARTag or ARToolKit Plus. However, we feel that the investigation of an easy-to-use ID-based extension to ARToolKit is justified due to the large number of legacy systems currently still in use. Although more sophisticated vision-based tracking libraries have been available for some time, conventional approaches like the ARToolKit are still frequently used thanks to their portability, regularly maintained code base, and the fact that a large number of users have experience in using them.

Acknowledgments

This work was supported by a Research Fellowship from the German Research Foundation (DFG).

References

[Azu97] AZUMA R. T.: A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments 6*, 4 (August 1997), 355–385.

[FEBS07] FISCHER J., EICHLER M., BARTZ D., STRASSER W.: A Hybrid Tracking Method for Surgical Augmented Reality. *Computers & Graphics 31*, 1 (February 2007), 39–52.

[Fia04] FIALA M.: *ARTag, An Improved Marker System Based on ARToolkit.* Technical Report, Publication Number: NRC: 47419, 2004, National Research Council Canada, 2004.

[FN03] FOXLIN E., NAIMARK L.: VIS-Tracker: A Wearable Vision-Inertial Self-Tracker. Proc. of IEEE Conference on Virtual Reality (VR), 2003.

[FNFB04] FISCHER J., NEFF M., FREUDENSTEIN D., BARTZ D.: Medical Augmented Reality based on Commercial Image Guided Surgery. In *Eurographics Symposium on Virtual Environments (EGVE 2004)* (June 2004), S. Coquillart and M. Göbel, (Ed.), pp. 83–86.

[IBJU*01] ISHII H., BEN-JOSEPH E., UNDERKOFFLER J., PIPER B., YEUNG L.: Urban Simulation and the Luminous Planning Table: Bridging the Gap between the Digital and the Tangible. *Journal of Planning in Education and Research 21* (2001), 195–202.

[KB99] KATO H., BILLINGHURST M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In *Proc. of the 2nd International Workshop on Augmented Reality (IWAR)* (San Francisco, CA, USA, October 1999), IEEE CS, pp. 85–94.

[NLL04] NILSEN T., LOOSER J., LINTON S.: Motivations for Augmented Reality Gaming. In *Proc. of Fuse (New Zealand Game Developers Conference)* (June 2004), pp. 86–93.

[PP04] PARK H., PARK J.-I.: Invisible Marker Tracking for AR. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)* (November 2004), pp. 272 – 273.

[RD06] REITMAYR G., DRUMMOND T.: Going Out: Robust Model-based Tracking for Outdoor Augmented Reality. In *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Santa Barbara, California, USA, 2006).

[TCD*00] THOMAS B., CLOSE B., DONOGHUE J., SQUIRES J., BONDI P. D., MORRIS M., PIEKARSKI W.: ARQuake: An Outdoor/Indoor Augmented Reality First Person Application. In *Proc. of International Symposium on Wearable Computers (ISWC)* (Atlanta, GA, USA, 2000), pp. 139–146.

[WS07] WAGNER D., SCHMALSTIEG D.: ARToolKitPlus for Pose Tracking on Mobile Devices. In *Proc. of 12th Computer Vision Winter Workshop (CVWW)* (2007).

[ZFN02] ZHANG X., FRONZ S., NAVAB N.: Visual Marker Detection and Decoding in AR Systems: A Comparative Study. In *Proc. of International Symposium on Mixed and Augmented Reality (ISMAR)* (2002), p. 97.