Eurographics Symposium on Virtual Environments (2004)
S. Coquillart, M. Göbel (Editors)

Authoring of Mixed Reality Applications including Multi-Marker Calibration for Mobile Devices

Jürgen Zauner† and Michael Haller‡

Upper Austria University of Applied Sciences, Media Technologies and Design, Austria

Abstract

Creative and innovative people have good ideas for new kinds of Mixed Reality applications. Applications designed by artists for example enrich the exhibitions of modern museums. Developing such an MR application is a complex task, which nowadays is realized by software engineers. We have developed an authoring tool, which integrates a user-friendly and intuitive calibration tool for developing MR applications.

1. Introduction

MR technology has been exploited in the medical, military, and most of all in the entertainment field. The use of Mixed Reality enhances users' perception and their interaction with the real world. It allows new ways for interaction and communication between the visitors and the objects of an exhibition. In the Virtual Showcase project, for example, the visitors get a 3D graphical augmentation of real objects placed inside a glass housing [BFSE01]. Most of the AR installations are not exclusively authored by artists, but programmed or scripted by programming experts like computer scientists. Moreover, an apparently small modification of the existing AR installation can cause a lot of modification tasks during the application setup. For instance the change of the tracking setup yields a dramatic modification of the existing code. Besides, a lot of creative and innovative people are not able to implement their brilliant ideas due to the lack of programming skills. An authoring tool solves this problem, because it enables people to produce new applications or maintain existing applications very fast and with a minimum of initial training. Indeed, artists and designers can directly work with the application and experiment with the new AR technology without the help of computer scientists. Furthermore, designers are often involved in virtual prototyping (cf. figure **??**). In most cases they need the help of programmers if they want to change the AR setup. In fact, a direct manipulation of the AR application seems to be an unsolvable problem.

Figure 1: *Artists don't only want to use an AR application, but they want to modify it personally.*

But, artists should be able to modify and tweak different application properties (e.g. change the surface properties of a virtual object or even change the GUI to improve usability). In other words, artists and designers should actively manipulate their AR applications already at the beginning and they should be able to do this without the need of experts. We believe that artists are a good example of end-users that have great ideas, but they are often hindered to express them, because of their missing expertise in programming.

† jzauner@fh-hagenberg.at
‡ haller@fh-hagenberg.at

2. Related work

A lot of authoring tools can be found in game industry: each game comes on the market including its own authoring tool, so that players are able to create their own worlds without a lot of skills. Few commercial tools are also available in the VR field (e.g. Virtools, EON Studio). Most of the current AR installations are either based on self-made frameworks (cf. ARVIKA, STAR, etc.), or on open source frameworks (e.g. Studierstube) [WT00, SFH*02]. The DART project is a collection of extensions for the Macromedia Director multimedia-programming environment, and therefore primarily designed for designers who want to develop their own AR applications [MMB*03]. Another approach is postulated by the AMIRE project, where the authoring tools are based on a component oriented approach but not on an existing multimedia authoring tool like Director [AMI02]. Together with the component oriented approach, AMIRE offers different authoring tools to support end-users during their setup process for an MR application. One of these tools, called CATOMIR, is described in the next section.

3. The generic authoring tool

We have developed the generic authoring tool called CATOMIR (Component oriented Authoring Tool for Mixed Reality). It is based on the component-oriented framework of AMIRE, which is the foundation of our applications that are authored with CATOMIR. We have analyzed the authoring process of applications and encountered three basic features for an authoring tool:

- Creating new components and configuring their initial behavior.
- Describing the workflow by connecting components (routing).
- Easily calibrating the placement of all virtual content that augments the real scene.

Authors analyze the application they want to build. During this analysis a set of required components will be figured out. So, the author has to take a look at the available components. If a component is missing, it can be loaded dynamically into CATOMIR. As soon as all components are available, the author creates each new component instance by choosing it out of a list of so called component prototypes. The use of categories enables the author to lookup the desired component very easily. Afterwards, the initial behavior is configured by a generic property editor. It uses the type information of each property to provide the adequate editor for the property. The workflow description is done by a so called connection editor. This editor allows the author to connect in- and out-slots of a component (cf. GraphEdit of DirectX). Each slot is strongly typed. Hence, only slots of the same type can be connected.

A component schema will become a very complex structure, which may not fit into the visible area. By a simple click with the stylus onto the component gui element 2D representation all other components and connections except the near neighborhood are blended out. On the left side of figure 1 the whole component schema of a MR assembly instructor program (cf. [ZHBH03]) is shown and on the right side a detailed view of a single component is given. Consequently, a more clear overview is achieved when less components are shown.

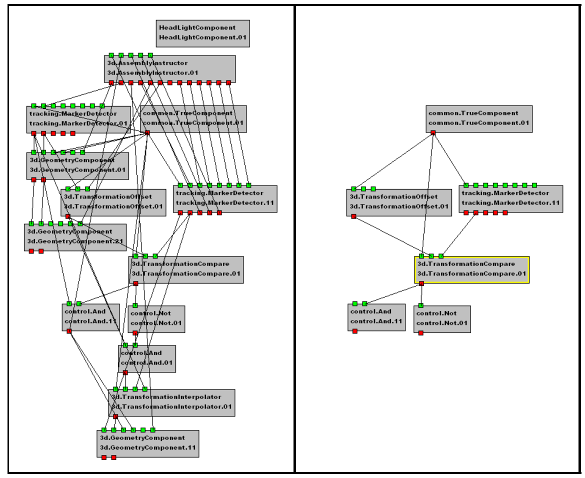

Figure 2: *View of a component's direct neighborhood only.*

4. Multi-marker detection

A lot of researchers use the ARToolKit library for tracking, because it is easy to integrate and the source is available. However, a marker based tracking system becomes problematic when larger areas have to be tracked. One marker is not enough to track a large area, because the marker can get out of the camera's field of view. Moreover, the marker can even be occluded when an object gets between the camera and the marker.

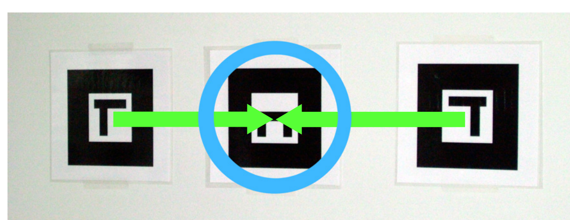

Figure 3: *All markers of a multi-marker refer to the circled highlighted reference-marker.*

The prevalent solution for this limitation of marker detection is to use several markers (as postulated by [BNR02]). This set of markers is called multi-marker. It is defined by the pattern and size of each marker and a transformation between this marker and a reference point. Each detected marker refers to this single reference point. We have decided to use the center of the first marker of the marker set as the reference point (depicted in figure 2). The arrows represent the transformations between the detected markers and the so called reference-marker.

Detecting a multi-marker can be done in several ways. The easiest way is to take the marker that has been detected best. Therefore, ARToolKit provides a so called confidence-factor. We just take the information of the marker with the highest confidence-factor. However, a better solution is to consider also all other detected markers of the multi-marker, because the position and orientation of each detected marker contain small inaccuracies. In the next subsection of this paper we present a method to get the average position and orientation of all detected markers. This method reduces the position and orientation error and, combined with a linear regression of the measured data, we are able to provide very stable multi-marker tracking.

4.1. Average filter

For each detected marker we get its position and orientation relative to the camera as a single four-by-four matrix. So, we have to extract a position vector from the matrix and a quaternion that represents the orientation. We do this task for each marker and get the sets of positions and orientations. Furthermore, we can calculate the average position vector and the average orientation quaternion.

Unfortunately, outliers in the orientation set of the detected markers have been observed. If the outliers' percentage gets too high then the tracking becomes instable until the outliers' percentage is lower. Using the statistical mean value instead of the statistical average value reduces the effect of outliers.

4.2. Linear regression filter

Better tracking results are achieved by using an average filter. However, we also want to improve the results for only one marker or stabilize the result of a multi-marker even more. Therefore, we have to consider the history of a tracking system and use a filter on a set of previous positions and orientations. We are using a linear regression on a small time period of the latest marker history. The linear regression provides us with two straight lines representing the correlation between the past time and the tracked placement (position and orientation). The calculated placement at the actual time is a good approximation of the real actual placement, because using linear regression allows us to minimize the average error for the placement during this small time period.

The tracking of a marker was quite stable at the idle placement of the marker and during the movement of the marker. We have observed an overshooting of the placement when the marker is moving and when it stops suddenly. This overshooting gets higher when the considered time period gets increased. By decreasing this time period we reduce the overshooting. But this is a tradeoff that is similar to the average filter approach.

So, our next approach was to reduce overshooting by taking the placement value of an earlier time on the actual calculated placement lines instead of taking the value of the actual time. Furthermore, we consider the confidence-factor of ARToolKit for the linear regression to reduce the influence of markers that are poorly detected. By using a weighted average between the calculated placement and the placement detected by ARToolKit we are able to parameterize the influence of the currently measured placement.

This final solution provides a very stable tracking of markers, especially in combination with multi-marker detection. However, the parameters of the system have to be adapted to the application scenario.

5. Multi-marker calibration

In the previous section we have described what multi-marker detection is and how it works. As we have already mentioned, the multi-marker detection needs transformations between each marker and the reference-marker. Authoring this manually would be very unhandy. So, we have developed a multi-marker calibration tool, which is integrated into CATOMIR. This tool detects all individual markers of a multi-marker. If the reference-marker is detected, the tool calculates and stores the transformations between all detected markers and the reference-marker. Further, it labels the detected markers as calibrated. Afterwards, it can use the already calibrated markers to calculate the transformations between uncalibrated markers and the reference-marker. This procedure is repeated until all markers are calibrated. To stabilize the transformations, the tool also recalibrates the calibrated markers by building an average transformation of all calibration steps.

The calibration steps have to be explicitly activated by the author, because using each frame for the calibration would increase the weight of the camera's positions where the author stays longer. This may lead to calibration errors. Hence, we have decided to let the author choose all calibration frames. Figure 3 shows an author calibrating a multi-marker on a Tablet-PC.

6. Placement of virtual content

In the previous section we have described the calibration of a multi-marker. But this calibration does not enable individual placement of virtual 3D content. This MR placement tool uses a placement-marker as a tangible input device and it supports four modes. The observation mode allows the author to take a look at the actual placement. In the second

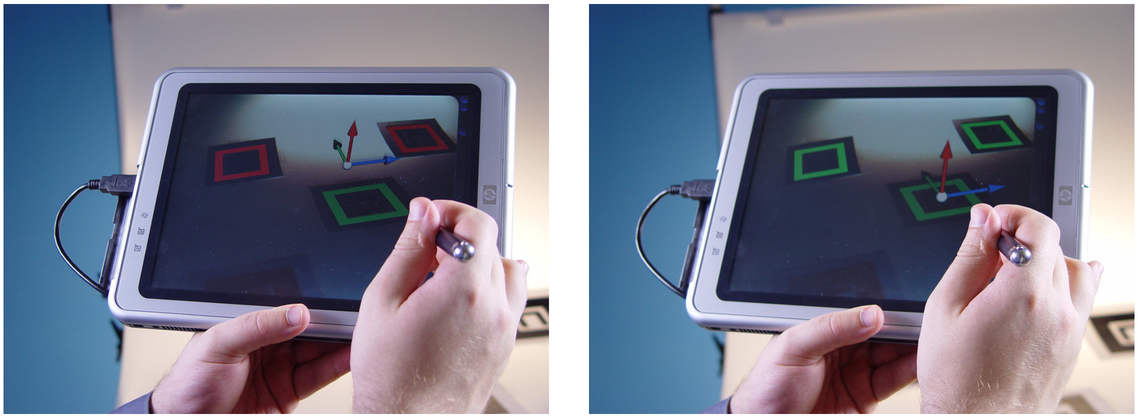

Figure 4: *The author calibrates the multi-marker on a Tablet-PC.*

mode, the author is able to modify the size of the object. The scaling factor is proportional to the distance changes between the reference-point (e.g. the shown reference-marker) and the placement-marker. The object is scaled down by moving the placement-marker near to the reference-point and scaled up by moving the placement-marker away from the reference-point. The third mode is used to move the object. The position changes of the placement-marker are applied to the object's position. Finally, the rotation mode enables the author to rotate the object. The orientation changes of the placement-marker are applied to the object's orientation. Separating the position calibration from the orientation calibration allows the author to modify the placement more accurately. Furthermore, the author is able to lock the modification on a specific axis. This is especially useful for the rotation of an object, because the object often has to be rotated by only one axis.

7. Conclusions and future work

We have shown that authors (especially designers) are able to create new MR applications with a generic authoring tool like CATOMIR. This is even true when the author has no experiences in programming. Without the calibration features of CATOMIR the calibration would be a very complex and time consuming task. Sometimes, this calibration task could take even longer than the description of the behavior. So, this is a very useful improvement for the authoring process of Mixed Reality applications.

References

[AMI02] AMIRE CONSORTIUM: AMIRE - Authoring Mixed Reality, 2002. http://www.amire.net. 1

[BFSE01] BIMBER O., FRÖHLICH B., SCHMALSTIEG D., ENCARNAÇÃO L. M.: The virtual showcase. *IEEE Comput. Graph. Appl. 21*, 6 (2001), 48–55. 1

[BNR02] BARATOFF G., NEUBECK A., REGENBRECHT H.: Interactive multi-marker calibration for augmented reality applications. In *IEEE / ACM International Symposium on Mixed and Augmented Reality, ISMAR 2002* (2002), IEEE Computer Society. 2

[MMB*03] MACINTYRE B., MARIBETH G., BOLTER J. D., DOW S., HANNIGAN B.: Dart: The designer's augmented reality toolkit. In *IEEE / ACM International Symposium on Mixed and Augmented Reality, ISMAR 2003* (2003), IEEE Computer Society, pp. 329–330. 1

[SFH*02] SCHMALSTIEG D., FUHRMANN A., HESINA G., SZALAVARI Z., ENCARNAÇÃO L. M., GERVAUTZ M., PURGATHOFER W.: The Studierstube augmented reality project. *Presence: Teleoper. Virtual Environ. 11*, 1 (2002), 33–54. 1

[WT00] WOHLGEMUTH W., TRIEBFUERST G.: Arvika: augmented reality for development, production and service. In *Proceedings of DARE 2000 on Designing augmented reality environments* (2000), ACM Press, pp. 151–152. 1

[ZHBH03] ZAUNER J., HALLER M., BRANDL A., HARTMANN W.: Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality* (Tokyo, Japan, October 2003).