Eurographics Italian Chapter Conference (2006)
G. Gallo and S. Battiato and F. Stanco (Editors)

VTK4AR: An Object Oriented Framework for Scientific Visualization of CAE Data in Augmented Reality

F. Bruno[1], F. Caurso[1], F. Ferrise[2], M. Muzzupappa[1]

[1] Università della Calabria – Dipartimento di Meccanica, Rende (CS), Italy
[2] Politecnico di Milano – Dipartimento di Meccanica, Milano, Italy

Abstract

In the last ten years many Augmented Reality (AR) applications for Scientific Visualization have been developed, attesting the effectiveness of this technique for data visualization and interaction. In all these applications, a software framework for scientific visualization was used to process data to be visualized, while an AR system was employed to display these data within an AR context. Hence, everyone who intended to approach the development of such applications should become necessarily familiar with the scientific visualization framework and the augmented reality one. This is of course an hurdle for the applications development, and the idea behind this work is exactly to provide a software framework that simplifies the development of such applications. With this in mind, we extended an existing and powerful open source library for scientific visualization (VTK) with few but useful classes for the interfacing with an existing AR library (ARToolKit) to easily handle the video see-through and the video-tracking functionalities. The resulting software tool, called VTK4AR, can be considered as an all in one software framework specific for scientific visualization in AR. Moreover, since it is built on top of VTK, it will be possible to employ a wide range of visualization techniques in many application fields. In particular, it has been tested in two AR applications: one for displaying data relative to a CFD simulation of a flow past a helmet, and another for displaying the forming error obtained prototyping an ankle support with the incremental forming technique.

Categories and Subject Descriptors (according to ACM CCS):
I.3.7 Computer Graphics---Virtual Reality; J.2 Computer Applications---Engineering; H.5.1 Information-Interfaces and Presentation---Artificial, augmented, and virtual realities.

1. Introduction

Recently, Augmented Reality (AR) has been successfully employed in scientific visualization tasks. In medical field, it has been employed for the visualization of CT (Computed Tomography), MRI (Magnetic Resonance Imaging) or ultrasound scans. In surgery planning, the required view of the internal anatomy is projected over the target surface to help the surgical procedure [GEK*98]. In ultrasound imaging it is e.g. possible to display the volumetric rendered image of the fetus overlaid on the abdomen of the pregnant woman. The image appears as if it were inside the abdomen and it is correctly rendered as the user moves [ACT*94], [SKB*01]. In these works, scientific data are initially obtained by CT or MR scans, then reconstructed, processed and visualized with specific viewers running on graphic workstations, and finally mixed with the live video images in accord with the tracking information.

Other applications of scientific visualization in AR have been developed upon Studierstube, a framework for collaborative AR [SSF*98], [SFH*02], [FLS*98], [BFS*03]. These applications allow users to display and examine volumetric data in a collaborative AR environment, and to highlight or explore slices of the volume by manipulating an optical marker as a cutting plane interaction device. In these works, instead, scientific data are obtained using a library for scientific visualization (AVS), then converted into the OpenInventor format, and finally visualized through the Studierstube framework.

From the papers mentioned above, it can be inferred that AR visualization is a powerful mean to improve the comprehension and the interaction of scientific data. However, the development of AR applications for scientific visualization is not immediate. Developers, in fact, have to master the scientific visualization framework and the AR system. Moreover, they also have to make them dialogue in the right way, since in all the previous works there is a total decoupling between this two frameworks. This decoupling is inevitable, but represents also an hurdle for everyone who want to develop such applications. Therefore, we thought to extend an existing scientific visualization library with some classes acting as interface with an existing AR library. VTK is the library employed for the scientific visualization, while ARToolKit has been employed for the video-tracking tasks. Both libraries (VTK and ARToolKit), are open source, hence it has been possible extend them to achieve the desired targets. The resulting framework, called VTK4AR exactly to highlight the dependence with the previous mentioned libraries, allows the developers to implement an AR application like a standard VTK application. In order to test VTK4AR, two applications have been implemented [BCM*05]: one for displaying data relative to a CFD simulation of a flow past a helmet, and another for displaying the forming error obtained prototyping an ankle support with the incremental forming technique [FAM*04]. In both, a TabletPC has been used to explore CAE data.

As it is easy to understand, video-tracking and AR visualization are performed by a single machine, since they are executed by a single application. Surely, this could represent a great problem in case of complex or huge data to be visualized, since in VTK4AR no parallel visualization has been implemented.

2. Hardware set-up

An augmented reality system can be implemented using different hardware configurations. One of the most widespread solutions is the optical see-through Head Mounted Display (HMD). Another chance to show digital data on a real scene is with the video see-through approach. One or two cameras can be employed to transform a normal HMD, a hand-held device [WS03] or a flat monitor [Rek97] into a simulated see-through display. Our system is based on a tablet PC equipped with a video camera that is employed both to simulate the see-through functionality and to track the display position with a pattern recognition algorithm (Figure 1).

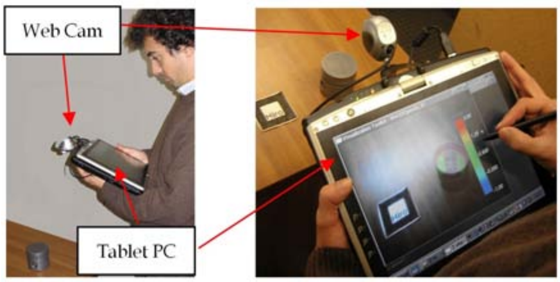

Figure 1: *Hardware set-up*

The use of the tablet PC as display device offers some advantages compared to other possible configurations. Compared to the HMD it is completely wireless and can be observed by two or three people at the same time, but, on the other hand it cannot support stereoscopic vision. Compared to a PDA it offers a wider display and much better hardware performances, but it weighs more. The video camera has been mounted on the top of the tablet, so it is possible for the user to handle the tablet with one arm, while the other arm holds the pen used to interact with 2D widgets displayed on the tablet monitor. The tablet works like a magnifying glass [RN95] that can be used to perceive information about the objects that, otherwise, could not be seen. The tablet can be imagined as an augmenting window that augments the perception of the objects visualized through it.

3. The occlusion problem

The central idea in all the AR applications is rendering the virtual objects above the real scene. In a video see-through AR application a webcam perceives the real world, and virtual objects, instead, will be drawn above this image accordingly to the position and orientation retrieved by the tracker, with the right projection parameters. So, like most AR applications, our AR Window will be made up by two overlapped viewports: one for displaying the real scene and another one for displaying virtual objects.

One of the problems in AR visualization is rendering virtual objects taking into account the objects that may occlude them. Although this problem may not be relevant for such applications, in scientific visualization the occlusion plays an important role. Consider the CFD analysis of a cylinder shown in figure 2.a. If the streamlines are drawn superimposed on the real scene the result is not realistic, because the streamlines "do not wrap" the cylinder. In figure 2.b the correct occlusion of the cylinder has been calculated. The data-set, employed in the example represented in figure 2, is a demo included in the VTK package.

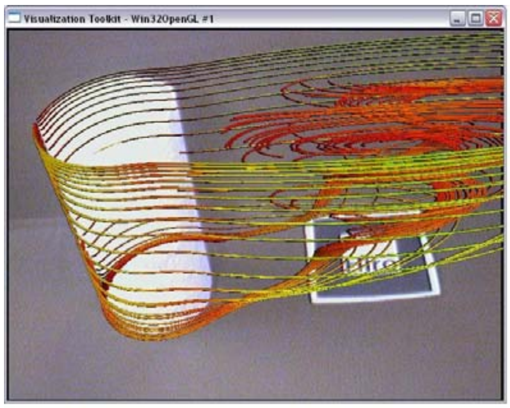

Figure 2 a: *Streamlines wrapping a cylinder without occlusion taken into account*

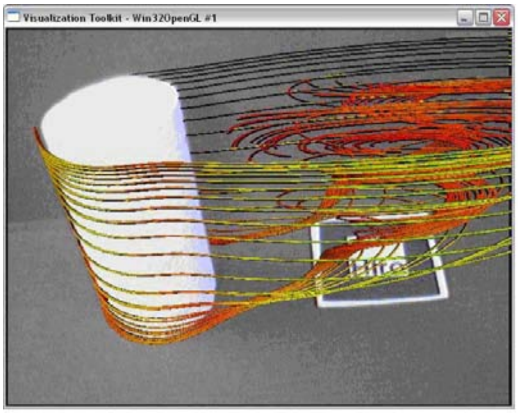

Figure 2 b: *Streamlines wrapping a cylinder with occlusion taken into account*

The occlusion problem has been tackled with different methods, but two general approaches can be identified. The first approach, defined as depth-based, assumes that geometries of the real objects are unknown. In the system presented in [BWR*96] the video output of the graphics workstation is combined with the output of a video camera using luminance keying. If the geometric models of the real objects are unknown, the depth-based approach can be employed [FHF*99], [Ber97], [LB00], [WA95]. Depth information have to be written in the Z-Buffer before rendering of the virtual objects occurs. With this approach the model of the real world is acquired reconstructing the depth of the real environment from the point of view of the camera. The other approach, defined model-based, implies that geometries of real objects are known. It means that the model of the real object, correctly positioned in the virtual world, can be used to produce occlusions by drawing it in black. These models have to be placed in the correct position in camera coordinates to produce an image identical to the live video image of the real ojects. In other words, when the geometric model is rendered, it appears at the same location and orientation in the computer-generated image as the real object in the live video.

Since in the visualization of CAE analysis, the geometrical models of the real objects are available, in VTK4AR the occlusion is taken into account with a model-based method. So to obtain a correct rendering it's sufficient draw in the Z-Buffer only the occluding objects, and finally render virtual objects. For drawing in the Z-Buffer only, VTK has been extended implementing a new VTK object, called vtkARDepthActor.

4.VTK4AR software architecture

The basis idea of this work is the development of an all in one framework specific for the Scientific Visualization in AR, provided with high level classes for the rapid development of such applications. In the development of these new classes we strived for a feeling of unit with VTK and for keeping invisible the interfacing with the ARToolKit library.

VTK4AR is totally object oriented, thus, since ARToolKit is a C library, the first step in the development has been the implementation of ARToolKitHandle, a C++ wrapping library to handle ARToolKit, that will be described in detail later. Thereafter, classes derived by some VTK base classes have been implemented. These ones provide:

- a special window in which render AR scenes
- an object for render the video-camera grabbed images
- an object for render 3D models in the Z-Buffer only, for the occlusion handling

The resulting architecture could be represented with the Figure 3, in which is highlighted that VTK4AR dialogues and interfaces itself with ARToolKit through the ARToolKitHandle library.

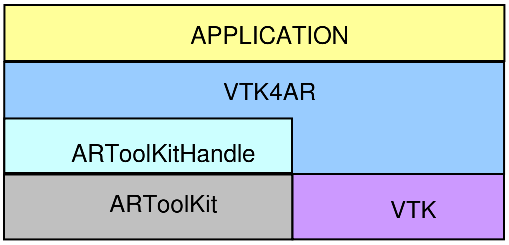

Figure 3: *Software modules layout of VTK4AR*

5. Implementation details

5.1 The ARToolKitHandle library

In order to have an easy handling of tracking and video libraries, and to have a full Object Oriented environment, we have developed a C++ API for managing ARToolKit C libraries (ARToolKitHandle). It mainly consists of two classes, one for video handling and one for tracking system handling.

All these classes are linked with the "original" ARToolKit libraries: all the methods, in fact, make only function calls to the ARToolKit libraries functions. This implies that it will be possible to employ the newest release of ARToolKit without changing the ARToolKitHandle code. The ARToolKitHandle library is made up of two main classes:

- ARVideoHandle, which provides an easy interface to the video devices, and it's quite simple to initialize and use.

- ARVideoTracker, that is an abstract class, implemented in ARMonoMarkerTracker and ARMultiMarkerTracker classes, for managing mono-marker tracking and multi-marker tracking. Every ARVideoTracker object has an ARVideoHandle object associated, from which it acquires images that will be processed.

5.2 New VTK classes

The first step in the implementation has been the development of the vtkARWindow class, directly derived by the vtkRenderWindow class. The main feature is that this class is made up of two overlapped viewports: one for rendering images grabbed by the video-camera (in background), and another for rendering 3D models (in foreground). In an vtkARWindow object it is possible to set directly the video-image obtained by an ARVideoHandle object in background through an apposite method.

In VTK an high abstraction level viewport called vtkRenderer is implemented. The new class implemented, vtkARImageRenderer, is derived from the vtkRenderer class, and its only task is to render the images grabbed by the video-camera.

Finally, the vtkARDepthActor has been implemented to achieve a correct occlusion handling.

This object is directly derived from the vtkOpenGLActor class. The main feature is that this object masks color buffers (by the OpenGL function glColorMask) before rendering. So it does not write in the color buffers but it only write in the Z-Buffer the depth-map of the occluding objects. Hence, the virtual objects will be rendered, only where they aren't occluded (i.e. when the Z of the virtual objects is lesser than the real ones).

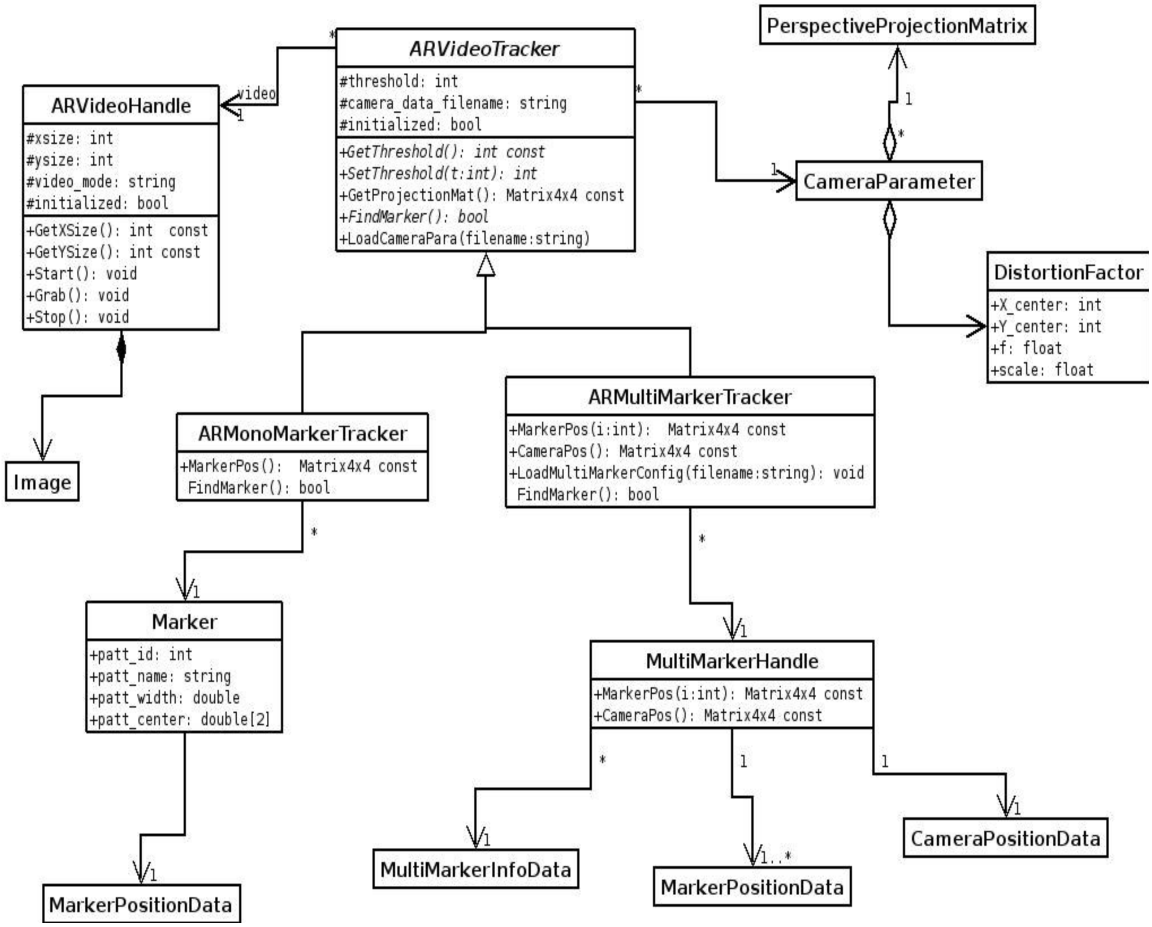

Figure 4: *UML class diagram of ARToolKitHandle library*

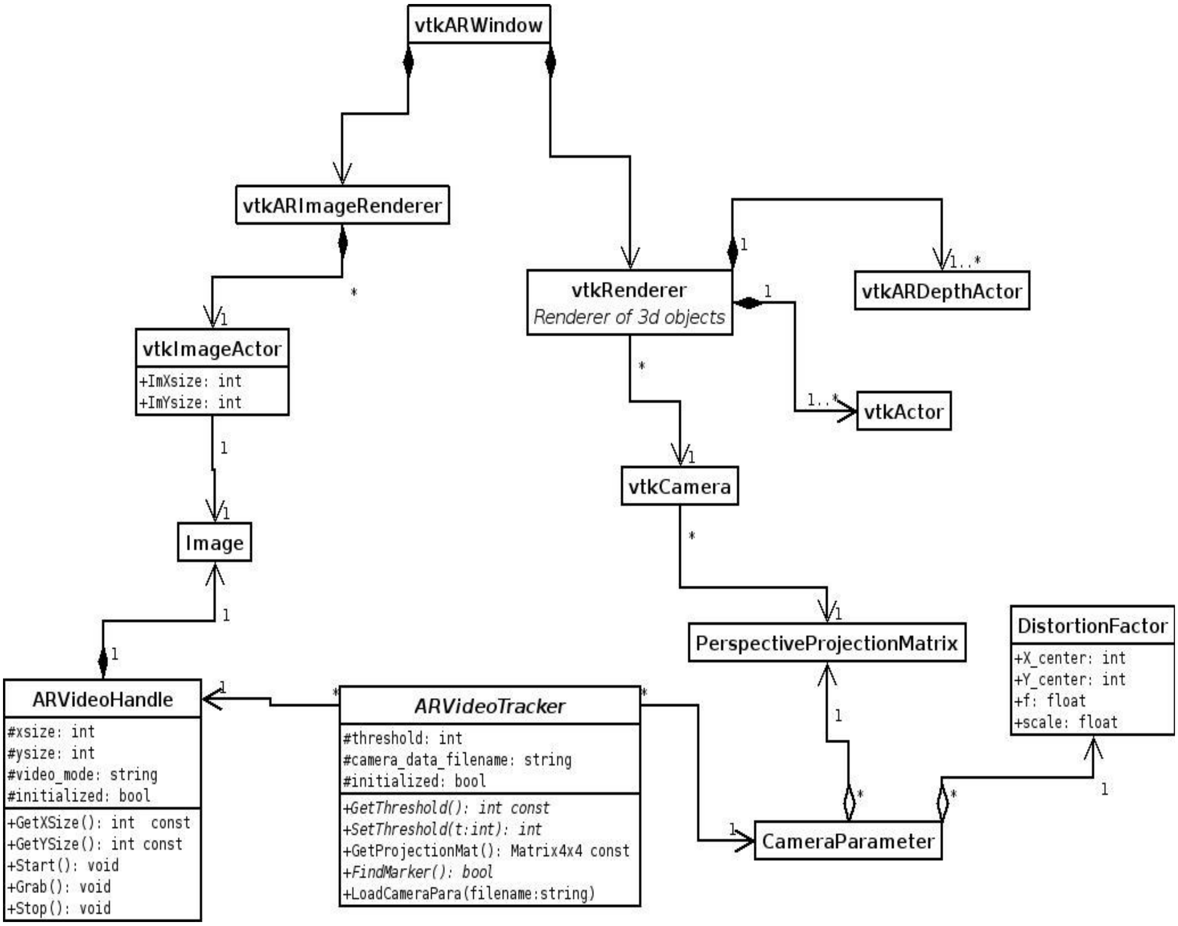

Figure 5: *UML class diagram of the framework*

6. Testing applications and results

Two applications of VTK4AR have been implemented in order to do some preliminary tests necessary to find out problems and advantages related to the visualization in AR. The first application regards the visualization of a CFD analysis carried out on a helmet (Figures 7, 8). The helmet has been acquired using reverse engineering techniques. Once the CAD model has been reconstructed, it has been analyzed with a CFD simulation software (Cosmos/FloWorks), and then the analysis data has been imported in VTK by a specific data reader developed for this purpose.

This example puts in evidence the importance of a correct calculation of the occlusion. Since the streamlines wrap the helmet, a coherent occlusion gives a great contribution to the realism of the visualization and plays an important role in the user perception of the data topology. Two different visualization techniques have been tested: streamlines (figure 7) and stream-surfaces (Figure 8).

Model	Number of points	Number of Polygons	Frame Rate
Stream-surfaces	12270	532	~28fps
Helmet	1723	3442	
Stream-lines	29921	58828	~20fps
Helmet	1723	3442	

Figure 6: *Table of performances*

The performances of the system, obtained in this test case, are reported in the previous table.

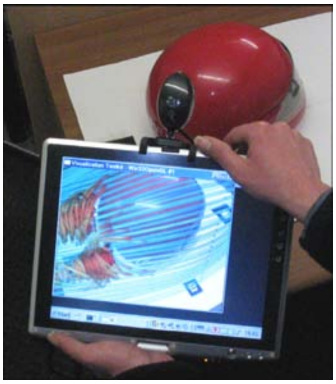

Figure 7: *Streamlines Visualization in AR*

The other test case shows discrepancy map between digital and physic mock-up of an ankle support, prototyped using Incremental Forming techniques [FAM*04]. In this kind of application, one of the main drawbacks is represented by the discrepancies between the desired shape and the obtained one. For this reason, a final measure of the manufactured product part has been carried out at the end of the production step using AR visualization.

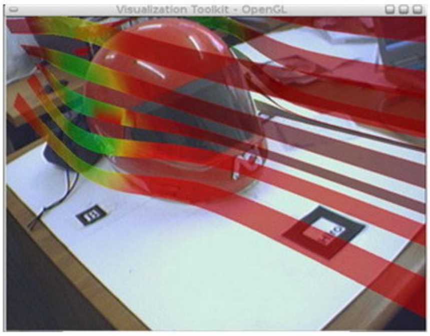

Figure 8: *Streamsurfaces Visualization in AR*

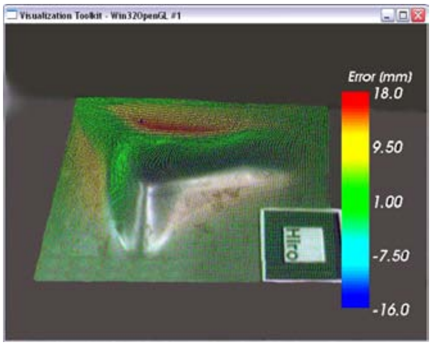

Figure 9: *Visualization of discrepancies map between the physical model and the CAD model of an ankle support*

7. Conclusion

VTK4AR allows to easily develop an AR application for scientific visualization, since the development is quite similar to a VTK application.

Further these preliminary tests have pointed out that the AR approach in the visualization of engineering data can give an intuitive 3D user interface that greatly simplifies data exploration and comprehension. Moreover it is a valid support in many decision making tasks since it allows collaborative work also with customers and/or not expert people, etc. On the other hand, visualizing data in VTK4AR requires data to be imported in VTK and some alignment of real and digital objects to be carried out. This task requires an effort that depends on the data complexity, on the scene extension in which the data are presented and on the number and the shape of physical objects that can occlude the data. But in some visualization tasks the use of augmented reality, not only improves the human-computer interaction. AR, in fact, is the straightforward way to compare numerical and experimental data and, also, allows the user to perform new kinds of analyses which would otherwise be impossible (i.e. some experimental analysis in which the results can not be expressed with numerical values).

The real potentialities of AR techniques in data visualization are still unknown and a software tool, that aims to simplify development of such applications, could be useful to investigate them.

References

[ACT*94] ANDREI, S., CHEN, D., TECTOR, C., BRANDT, A., CHEN, H., OHBUCHI, R., BAJURA, M., FUCHS, H. Case study: Observing a volume rendered fetus within a pregnant patient, Proc. IEEE Visualization, Washington, USA, (1994-10)

[BCM*05] BRUNO F., CARUSO F., DE NAPOLI L., MUZZUPAPPA M., Visualization of industrial engineering data in Augmented Reality, Journal of Visualization, submitted for publication

[BER97] BERGER, M.O. Resolving occlusion in augmented reality: a contour based approach without 3D reconstruction, Proc.IEEE Conference on Computer Vision and Pattern Recognition, (San Juan), Puerto Rico, (1997-6).

[BFS*03] BARAKONYI, I., FAHMY, T., SCHMALSTIEG, D., KOSINA, K. Collaborative Work with Volumetric Data Using Augmented Reality Videoconferencing, Proc. IEEE and ACM ISMAR, (Tokyo), (2003-10).

[BWR*96] BREEN, D.E., WHITAKER, R.T., ROSE, E., TUCERYAN, M. Interactive Occlusion and Automatic

Object Placement for Augmented Reality, Computer Graphics Forum, 15-3, (1996).

[GEK*98] GRIMSON, W., ETTINGER, G., KAPUR, T., LEVENTON, M., WELLS, W., KIKINIS, R. Utilizing segmented MRI data in image guided surgery, International Journal of Pattern Recognition and Artificial Intelligence, 11-8, (1998), 1367-1397.

[FAM*04] FILICE L., AMBROGIO G., MUZZUPAPPA M., DE NAPOLI L., FRATINI L., COSTANTINO I., Influence of some relevant processparameters on the dimensional accuracy in incremental forming: a numerical and experimental investigation.Journal of Materials Processing Technology, 153-1, (2004), 501-507.

[LB00]LEPETIT, V., BERGER, M.O. Handling Occlusion in Augmented Reality Systems: A Semi-Automatic Method, Proc. IEEEand ACM International Symposium on Augmented Reality, Munich, Germany, (2000-10).

[SFH*02]SCHMALSTIEG, D., FUHRMANN, A., HESINA, G., SZALAVARI, ZS., ENCARNAÇÃO, L. M., GERVAUTZ, M., PURGATHOFER, W. The Studierstube Augmented Reality Project PRESENCE - Teleoperators and Virtual Environments, 11- 1, (2002-2)32-54.

[SKB*01] SAUER, F., KHAMENE, A., BASCLE, B., SCHIMMANG, L., WENZEL, F. AND VOGT, S. Augmented reality visualization of ultrasound images: System description, calibration and features, Proc. IEEE and ACM International Symposium on Augmented Reality, New York, USA, (2001-10).

[SSF*98] SZALAVARI, ZS., SCHMALSTIEG, D., FUHRMANN, A., GERVAUTZ, M. STUDIERSTUBE - An Environment for Collaboration in Augmented Reality, Virtual Reality - Systems, Development and Applications, 3-1, (1998), 37-49.

[FLS*98] FUHRMANN, A., LÖFFELMANN, H., SCHMALSTIEG, D., GERVAUTZ, M. Collaborative Visualization in Augmented Reality, IEEEComputer Graphics & Applications, 18-4, (1998) 54-59.

[FHF*99] FUHRMANN, A., HESIAN, G., FAURE, F., GERVAUTZ, M. Occlusion in collaborative augmented environments, Computersand graphics 23-6, (1999-12), 809-819.

[RN95] REKIMOTO, J. AND NAGAO, K. The world through the computer: Computer augmented interaction with real world environments. In Proc. ACM Symposium on User Interface Software and Technology, Virtual and Augmented Realities, pages 29–36, 1995.

[REK97] REKIMOTO, J. NaviCam: A Magnifying Glass Approach to Augmented Reality, Presence: Teleoperators and VirtualEnvironments, Vol. 6-4, (1997-8) 399-412.

[WA95] WLOKA, M. M., ANDERSON, B. G. Resolving occlusion in augmented reality, Proc. Symposium on Interactive 3D graphics,Monterey, California, 1995.

[WS03] WAGNER, D., SCHMALSTIEG, D. First Steps Towards Handheld Augmented Reality, Proc. 7th International Conference on Wearable Computers, White Plains, NY, USA, Oct. 21-23, (2003-10).