Projected Reality – Content Delivery Right onto Objects of Daily Life

Jochen Ehnes and Michitaka Hirose

***Abstract*—Spatial augmented reality using controllable projector-camera-systems to project onto objects directly, or Projected Reality as we call it, offers the possibility to augment objects without requiring their users to wear or carry any devices. In order to provide the freedom of movement to users, we developed an architecture that allows projected applications to roam a network of projection units to follow the users. The concept of connecting physical objects with content in form of projected applications, although devised for projection based augmented reality systems, can be applied to HMD based or hand held AR systems as well. After a description of our AR projection system and a example application that could be used to provide location specific information to travelers using public transportation systems, we will lay out our vision of a system that may be able to provide content on a global scale.**

***Index Terms*—Application Roaming, Multi Projection System, Projected Reality, Spatial Augmented Reality.**

I. INTRODUCTION

In order to support users performing different tasks without being a hinderance, we developed an Augmented Reality (AR) system based on a combination of a video projector and a video camera mounted on a pan and tilt device. We consider this as an alternative to wearable computers and Head Mounted Displays (HMD). Using this technology, we built a system that can project augmentations on fixed, as well as movable objects around the projection device. However, the nature of the projection introduces some limitations:

1) Objects have to be close enough to the AR-projection system so that the system can detect the markers. Also the resolution of the projection decreases with distance, which renders augmentations unreadable if the object is too far away.
2) Surfaces that shall be augmented have to face the projector since a projection is not possible otherwise. While certain angles between the surface normal and the direction of projection can be compensated by predistorting the projected augmentation, the quality of the projection clearly decreases with increasing angles. At 90 degrees a projection becomes impossible.
3) Finally, the line of sight between the AR-projection system and the object to be augmented must not be blocked. Otherwise the augmentation may be partially shadowed or, if an object's marker is not completely visible, no augmentation would be generated in the first place. Depending on the location of the projection system, users may block the line of sight easily[1].

In order to overcome these limitations we use several networked AR-projection units. We developed an architecture consisting of an application server and any number of projection units that allows our projected applications to roam these units.

While this architecture makes it possible to equip a room or building with projection units that can augment objects within that room or building, it could be extended to reach global scales. Such a system would not only be able to augment the same object in many places, it would open up new ways to deliver dynamic content with otherwise static media, such as newspapers, books and magazines. And that would be just the beginning as we believe. Although we believe that our architecture is a suitable way to provide that functionality, it does not seem to be possible to implement a system based on it with the current technology. Consequently, we will point out where the current (tracking) technology needs to be improved and what additional features would be necessary to implement such a global system.

II. PREVIOUS AND RELATED WORK

This work builds on our previous work which we published as follows. In [5] we introduced our hardware setup consisting of a video-projector mounted in a controllable pan and tilt device (AV4 from Active Vision) and a video camera mounted on top of the projector (see Fig. 1). We furthermore described our software that can project augmentations on fixed as well as on movable objects using this hardware. In order to ease the development of applications that augment objects using this projection system, we developed an Application Programming Interface (API), which we introduced in [6].

Manuscript Received on September 25, 2006.
Jochen Ehnes, The University of Tokyo
Michitaka Hirose, The University of Tokyo
E-mail: hirose@cyber.rcast.u-tokyo.ac.jp

[1]In order to minimize the blocking of the projection by the user, we suggest to place the projection systems high enough above users' heads, projecting down from the front at the most common position of the users. Furthermore, if the projection system always projects some light (grey background instead of black), it is also perceived as a light source and people instinctively place themselves in such a way that their work area is well lit, keeping the line of sight free.

Fig. 1. A projection unit consisting of a projector and camera on a pan and tilting device. See Color Plate 6.

In [4] we introduced the idea of using several similar AR-projection systems to extent the usable range, as well as to enable augmentations from all sides. Finally, in [3] we presented the results of some user tests about how humans perceive projection quality depending on projection distance and projection angle. Other controllable projector camera systems have been presented in [14] (although the application presented here doesn't seem to make use of the possibility to control the system's orientation), [13] and [1]. In [15] the usage of several I/O-Bulbs (projector/camera systems) has been described as well, however they did not provide an application independent architecture and they were not concerned about huge networks of projection units either.

There also have been several works on augmenting books with additional information. [11] describes a the augmentation of a physics text book to provide a more effective learning environment for students. It also describes an interactive Venn Diagram, which is aimed at supporting effective information retrieval. While these applications work on their 'Enhanced-Desk' environment (big parts of the augmentation and interaction happens on the desk), we believe it would be much more useful if the augmentation was not restricted to that desk. An other research group worked on the recreation of digital documents on real paper, both using video projection [9] and HMDs [7].

Finally, [2] describes a "Framework for Generic Inter-Application Interaction for 3D AR Environments". A mechanism as the "Data Flows" described therein could be useful to establish communication links between applications that augment physical objects in close proximity in order to let them work together or exchange information. However, this gets more complicated on a system where the applications roam different projection units.

III. PROJECTED APPLICATIONS AND APPLICATION ROAMING

The basic idea of application roaming in our network of AR-Projection units is as follows. The units themselves have no knowledge about how to augment different objects. This information is coded as 'Projected Applications', which are analogous to applications in a GUI environment. However, while conventional applications interact with the user via windows and widgets on a computer's screen, projected applications use tracked objects for interaction. The AR-Projection systems provide means to identify tracked objects as well as to measure their positions and orientations. They also project the output onto these tracked objects or other fixed and known surfaces, such as walls (encoded in the projected application). The AR-system can be seen as an operating system that loads and executes projected applications and provides an abstraction layer for these applications to communicate with the user and to control the hardware.

The projected applications reside on an application server. However, the application server not only serves the applications for the AR-projection systems. More importantly, it also maintains the state of the projected applications and enforces that the state is only modified by one AR-system at the time. Once a projection system detects a marker, it sends its ID to the application server. In reply, the application server sends the application[2] and, if available, the display rights and state of the application back to the projection system. Now the projection system starts the new application. If the system was granted the display rights and sent the last state of the application, it initializes the application with its state and starts to project the augmentations. If the display rights were not available at that moment, the system does nothing but trying to follow the object and waiting to be granted the display rights. If a system that owns the display rights for an application cannot detect the corresponding marker any longer, it returns the display rights together with the current state to the application server. It can reapply for the display rights once it detects the marker again. In the mean time, the application server may send the state and display rights to another projection system.

While this simple method of managing the display rights requires only minimal amounts of network bandwidth (see Fig. 2) and is sufficient if the ranges of the projection systems do not overlap, it is not satisfactory if more than one system could perform the augmentation at one time. When two systems compete for the display rights, the system that detected the marker first gets them and keeps them until the marker disappears from its camera's view. However, it would be better if the system with the best view of the object would project the augmentation. Furthermore, the management of the display rights should be more dynamic and find a new projection system that takes over before the active one looses the object and the augmentation disappears completely.

[2]In our prototype it currently only sends the application's name.

IV. OPTIMIZING THE QUALITY OF PROJECTION

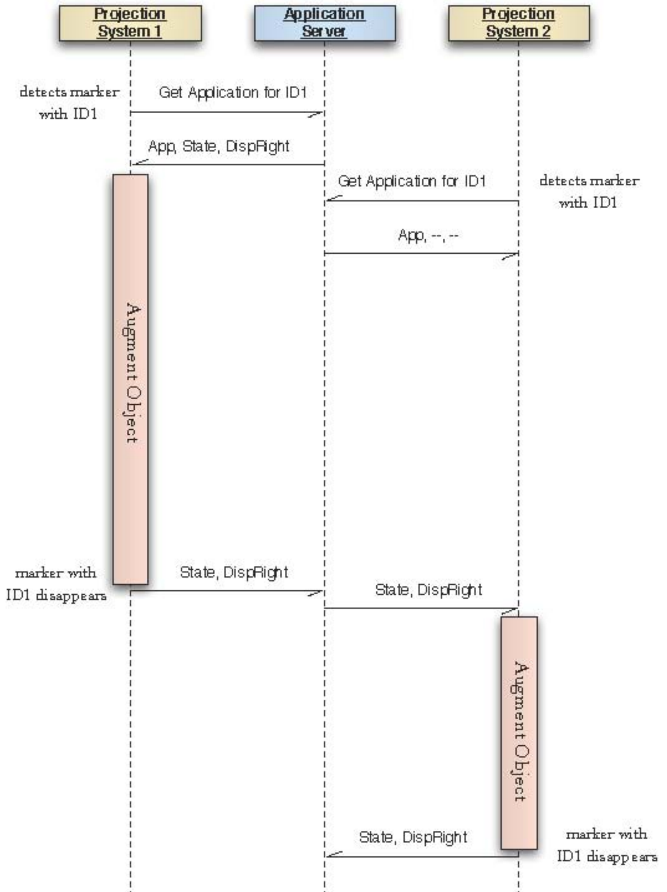


Fig. 2. The message flow for our simple method of application roaming.

In order to maximize the visible quality of the projected augmentation, as well as to make the transition between two projection systems as smooth as possible, we had to extend the management of the display rights and make it more dynamic. The application server can actively withdraw display rights (in combination with the applications' states) from a projection system now. This way it can give the display rights (and states) to better suited projection systems at any time, long before the augmentations disappear completely on the systems that held the display rights before. However, in order to decide which system gets the right to augment an object, the application server needs to know which system is suited best to do it. But what qualifies a system to be the best suitable? In the introduction we enumerated the three main points, (1) distance, (2) direction of the surface normal and (3) free line of sight. The quality can be considered to be a weighted combination of these criteria. However, the weighting may be quite task or application specific. In order to keep the task of the application server simple, we introduced a scalar quality value (Section IV-A). The system with the highest quality value is considered the best and consequently should perform the augmentation. Since the criteria of quality can be very task specific, we decided that the quality value is not something the system can provide. Therefore the projected applications must provide a function that calculates the quality value. The AR projection systems regularly send updates of the quality values of all applications they host to the application server. In consequence the application server can easily compare the quality values of the different applications running on the available AR projection systems and in turn can ensure that every application runs on the optimal projection system.

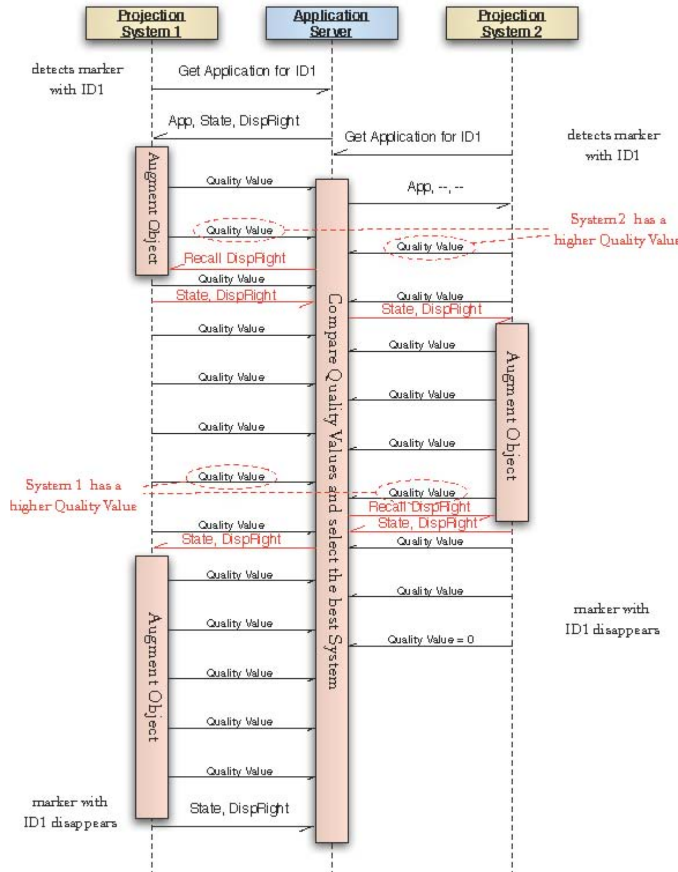


Fig. 3. The message flow for our advanced method of application roaming. The application server constantly compares the quality values sent by the different projection units in order to determine the best one.

A. *The "Quality Value"*

A crucial point in order to find the optimal projection system to perform the augmentation is the estimation[3] of the projection quality that each system could achieve. In [3] we examined how human test subjects evaluated the projection quality onto tracked objects depending on projection distance and angle, which lead to a formula to estimate the projection quality based on these factors. Furthermore we assume for simplicity that the projection is not shadowed if the camera can 'see' the marker on the object, as projection surface and marker, as well as projector and camera are located closely together and objects in the middle should block both the tracking and the projection.

[3]While it seems desirable to actually measure the quality of projection in order to be more accurate, it in fact would be more disturbing to the user than eventual errors in the estimation can be expected to be. To measure the quality of projection would mean that the systems would have to take turns in projecting the augmentation and then measuring the result. This switching between different systems would be very obvious, especially if the differences in the projection quality are huge. Consequently, a measurement of the projection quality is out of the question, unless one could perform it in a way invisible to the human eye, such as by using infra red (IR) or ultra violet (UV) light.

V. IMPLEMENTATION

A. Test Setup

Our test setup consists of three projection systems (AV4 from Active Vision with a Sony DFW-VL500 Firewire camera mounted on it, connected to a dual G5 Power Mac). For more details about the setup, control of the pan and tilt unit and the calibration of various parameters including the offset between camera and projector we would like to refer to [5]. In our current setup, as illustrated by Fig. 4, we control all three projection units with the same DMX512[4] bus. This way we need only one USB-to-DMX512 converter box. In order to be able to send values on the DMX512 bus from all three computers to control their projector, we implemented a server program that runs on the machine with the interface box connected to it. The AR application server runs as an independent application as well. Just as the DMX512 server, it may be running on one of the AR projection systems' computers.

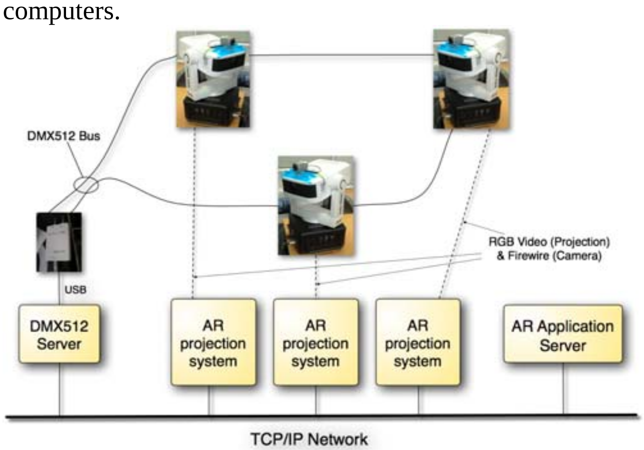
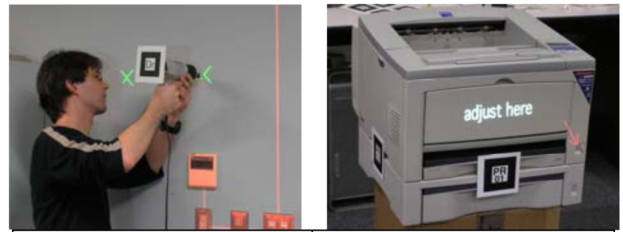


Fig. 4. The setup of our test system.

The projected applications are implemented as subclasses of our 'application' class (in Objective-C). We use the dynamic capabilities of the Objective-C runtime system to instantiate objects of these application-subclasses from their class names when needed. Currently we still link the object files implementing the applications (Objective-C classes) to the executable of the projection system, so the instantiation of these objects does not require any more actions. However, in future we plan to pack these application classes in bundles containing the code in form of a dynamically linked library and the necessary additional files like images for textures etc. These could be loaded from the application server before the application objects are instantiated.

B. Example Application

While developing the AR projection system as well as the architecture to enable applications to roam between several projection systems, we implemented several example applications. Our main purpose for developing them was to test the functionality and to explore the possibilities of the AR projection systems. We developed applications that augment the room itself (Fig. 5(a)) as well as objects within a room such as the printer in Fig. 5(b).



(a) "Drill Guide" shows where to drill holes and where not to in the vicinity of the tracked drill	(b) Machines like this printer an be augmented to help the user find the right control elements

Fig. 5. Augmenting a room or objects within the room. See Color Plate 7.

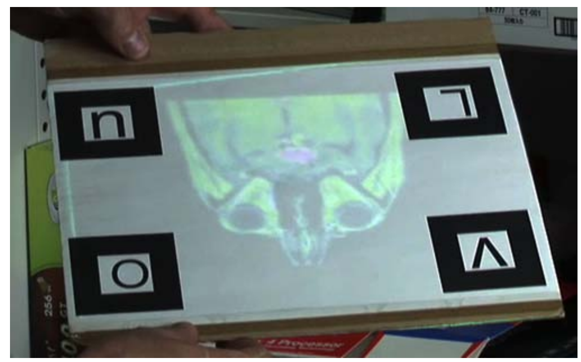


Fig. 6. A physical cutting plane through a virtual volume (part of a human head). See Color Plate 8.

We also experimented with less traditional AR applications, such as the interactive visualization of volume data sets. Fig. 6 shows a board augmented with the image of a cutting plane through the volume data set that coincides with the board.

However, the demo application that illustrates the benefit of a large scale network of projection systems best is our Guiding Ticket example:

1) Guiding Ticket: We envision the usage of AR projection systems in public spaces. If train stations for example were equipped with a sufficient number of projection systems, it would be possible to guide travelers to their connecting trains. Suppose a person wants to travel from A to B using public transportation and has to change several times on the way. If the traveler does not know the places to change, it can be quite stressful to find the right connection on time. With the system we envision, the traveler could get a special form of ticket that offers room for augmentations. Whenever the travelers have to change trains, they can take out the ticket and wait for an AR projection system to detect it. The system then projects on the ticket the information about the track the train is going to leave from, the wagon where the reserved seat is, how much time there is until the train leaves, as well as an arrow that always points in the direction to walk to. Fig. 7 shows the augmented ticket.

The text fields in our prototype application just display preset strings and a timer counting down towards a point in the near future. The direction of the arrow is calculated based on the X,Z coordinate of current position and the goal position. The goal position may depend on the current position as well, so that a chain of goal positions can be programmed to direct a user around obstacles. However, there has not been implemented any form of route planning. Since we only have

[4]DMX512 is the most common communications standard used by lighting and related stage equipment. A nice introduction can be found at http://www.dmx512-online.com

three projection units which we would set up in one room so that their ranges of projection overlap, it has not been possible to really test the application's usefulness to navigate between several locations. Nevertheless, the application provoked a lot of positive responses at public presentations.

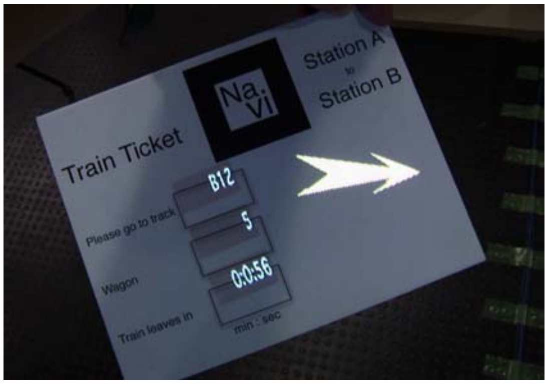


Fig. 7. Future train tickets may guide you to your next train.

VI. SCALING UP THE SYSTEM

For the future, as projectors and cameras get smaller and cheaper, we envision a network of projection units in many (public) places and application servers as common as web servers today. In such an environment information could be obtained by holding the corresponding objects (tickets, magazines, Books etc.) in front of any of those projection units. We already illustrated the possibility to augment special train tickets which could help anyone to find the connecting train fast, which do not require any knowledge about using computers. We believe that there are countless more applications, such as the augmentation of books, magazines or newspapers. While many people feel more comfortable reading printed media than reading from a computer screen, the online media is usually more up to date. This is inevitable, as the production and distribution of the print media requires a certain amount of time. A hybrid form of media could satisfy both. The printed part would be as comfortable to read as any other printed media, while it could be augmented with up to date information when in reach of an AR-projection system. Such a system could furthermore display movies on the paper instead of just images. However, in order to implement such a large distributed AR system, several changes would be necessary.

A. *Two Level Application Server Architecture*

It not only would be unnecessary to use the advanced application roaming algorithm described in Section IV to coordinate projection units far apart, it would also result in a lower quality due to the time lag introduced by connections to distant application servers and would increase the overall network load. In order to keep the network load low and the switching of projection units fast, we propose a two level application server architecture. All projection units in a room would be connected to a local application server (a proxy server), that performs the roaming based on quality values (Section IV). The local application server connects the room to the rest of the world and communicates with the application servers around the globe using the simpler protocol as described in Section III. Fig. 8 illustrates that.

B. *Enhanced Video Tracking*

The majority of possible and necessary improvements lies in the area of the video tracking system. As we have not been developing video tracking systems so far, the following points (at least the first and the third) can be seen as a wish list to the video tracking community.

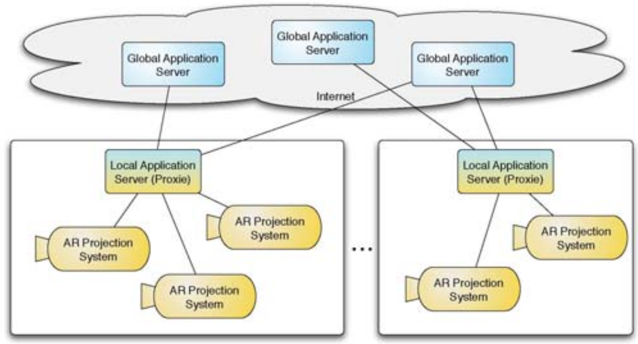


Fig. 8. Hierarchy of AR-Application-Servers for a network of AR Projection Systems on a global scale

1) URLs in Markers: Currently we are using the ARToolkit [10] tracking library to detect markers in the images captured by the camera. When a projection unit starts up and establishes its connection to the application server, it first downloads the marker patterns from there and adds them to the list of AR-Toolkit's known markers[5]. Only then a projection unit is able to recognize markers of tracked objects and request the PA for it. This however is not practical for a large number of application servers. In order to be able to set up a network of application servers and projection units on a global scale, it is necessary that the projection units can decode the application server's address and the ID of the application on that server from the marker. Using ARTag[6] instead of AR-Toolkit would be a step in the right direction in this regard, as it decodes the marker ID from the marker's pattern directly instead of using a pattern matching algorithm to identify the marker. However, with its limitation to 2048 markers, it would not be able to satisfy the needs to distinguish millions of objects. Neither would it be possible to encode a server's address using these eleven bits. One would have to extend ARTag in order to increase the number of bits encoded in the marker.

   On the other hand markers should be as small as possible, which limits the amount of information that can be encoded in them. A standard URL encoded as ASCII text seems to be much to big to be encoded in a marker. Such a marker would be so complex that it would have to cover a huge area of the camera image to enable the tracking system to decode the pattern. This would

[5]The projection unit already knows two markers that are used for calibration purposes.

[6]A marker tracking system by Mark Fiala. In contrast to AR-Toolkit it uses a line detection approach instead of a threshold, which makes it more robust to lighting changes. Furthermore it decodes the marker ID from the pattern directly and does not rely on pattern matching. It can be downloaded at http://www.cv.iit.nrc.ca/research/ar/artag/ .

seriously limit the range of the tracking system. However, if a marker could encode six bytes, it would be enough to encode the IP-address with four of them and to use the remaining two bytes to identify the application object on the server. This would allow a server to manage the augmentation of 65535 physical objects[7]. Adding another one or two bytes would allow for an even greater number of managed objects per server.

When books, newspapers or magazines are to be augmented, individual copies of the same book or paper would have to share the same markers. That way the number of individual markers needed, as well as the cost to print these objects can be reduced a lot. That means that individual copies can not have their own state and store user dependent data. At the same time that should rarely be necessary, as the augmentations are only showing new information provided by the publishers of the printed media[8]. However, if some user dependent data would have to be stored, one may introduce something like a virtual wallet, which may be used to store user specific data from other applications. If users bring such a wallet object into close proximity of the printed media, the two corresponding applications could start to communicate with each other. Then the information could be stored on the server that manages the virtual wallet. In a similar way the information could be retrieved later.

2) Invisible Markers: Markers have to be of a certain minimal size, so that the camera of a projection system an recognize them from a certain distance. Furthermore the accuracy of the tracking gets better with increasing marker size, or if several markers are combined into a multi-marker. In practice however that means that a large portion of the object's surface is covered with markers. Space that otherwise could be used to project augmentations on. Furthermore, it is important not to project anything onto the marker, since that results in a different visual appearance and consequently the marker can not be recognized any longer and the augmentation disappears. Of course, once the augmentation projected onto the marker disappears, the marker is recognized again. As a result the augmentation starts to flash. While it is possible to hide the marker in a video-see-through AR system, that is not possible with a projection based AR system. And if it was possible, it would result in the flashing effect described above. An approach to improve this situation is to use markers that are invisible to the human eye, but visible in another part of the spectrum, such as IR (in [12] an IR marker is used to visibly hide the marker) or UV. With such a marker it would be no problem to project an augmentation in the visible part of the spectrum over the marker. A video tracking system using a camera that can receive the marker's wavelength, equipped with a filter that blocks out the visible light, would not be disturbed by the projected augmentations either.

3) Non Planar Markers: For all marker based tracking systems we know of so far, it is a requirement that the markers are planar. Only then their position and orientation in 3D can be calculated. However, in order to be able to augment a newspaper, a magazine or a real train ticket, it would be necessary to track curved markers and measure their curvature as well. While it has been demonstrated before that a grid of AR-Toolkit markers can be used to find the $''d''\rightarrow''D''$ coordinates of their corners, and use these $''d''\rightarrow''D''$ coordinates to distort a texture of an image that gets overlaid onto the curved surface that way, it is only possible with video see through AR systems. The offset between projector and camera (as well as the offset between camera and eye in an optical see through HMD) makes it a necessity to have 3D coordinates. While [9] describes a system that projects virtual pages onto a book of empty pages, they require the projection of a lattice pattern to be projected onto the paper in order to calculate the homography necessary to warp the projected image. Since that has to be done every time the paper moves, we cannot consider it as a convenient solution from a user's point of view.

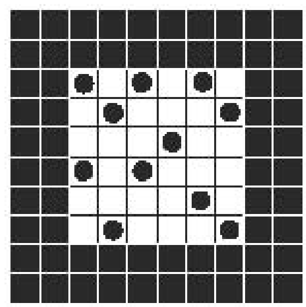


Fig. 9. A grid overlaid over a conventional marker could be used to detect the curvature of the surface.

While one certainly could try to calculate 3D coordinates, curvature etc. from a grid of conventional markers, it may be possible to modify the setup of the markers, so that this information could be calculated from one marker instead. For example a grid of lines could be overlaid over a square marker as illustrated in Fig. 9. That grid would provide the additional information needed to calculate the curvature of the surface if the marker is relatively close to the camera. If the marker is too far away that the camera can detect the grid, it still should be possible to detect the marker as if it was a conventional one. If a grid of conventional markers was used instead however, the individual markers would all be too small to be detected. This way a single non planar marker would have an implicit level of detail functionality. Marker ID, position, orientation and curvature could be detected when close enough, and only ID, position and orientation when further away. If the marker is so far away that the curvature can not be detected, it should be acceptable to draw the augmentation as a flat rectangle as well.

[7] That is if every object is identified by one marker only. If multi markers are to be used, the number of physical objects an application server can manage is reduced accordingly.

[8] Of course it is necessary to keep the position of the play head if the projection is going to be handed over to another projection unit while playing animations or movies. However, it should be possible to identify individual copies based on their locations. Such a temporal state could be kept by the local/proxy application server. However, once the user closes the book or paper and moves to another location, this kind of state would be lost.

VII. FUTURE WORK

Although the above section about an enhanced video tracking system could be considered future work as well, we do not plan to work on that for now. We plan to work on the following instead:

A. Objects with Several Surfaces

Currently Projected Applications roam projection systems as a whole. They run on at most one unit at a time and they can also project their augmentations only from that projection unit. That means that the objects they augment have to be flat. Otherwise just one face can be augmented reliably. We are currently looking into possibilities to distribute an application dynamically, such that different surfaces of an object can be augmented by different projection units independently, while keeping the state consistent. We try to do this in a way that keeps the necessary chores like networking and multithreading from the application developer as much as possible.

B. Advanced Shadow Removal

A system as described in the previous subsection would also allow us to experiment with more sophisticated ways of shadow removal by projecting the same augmentation with more than one projector at a time. While this principle has been introduced with the Virtual Rear Projection systems [8], it seems much more challenging when everything is moveable. Also the tracking jitter is currently so high that it seems hard to imagine to perform an invisible blending of an augmentation projected by two projection units though.

REFERENCES

[1] Andreas Butz, Michael Schneider and Mira Spassova. Searchlight - alightweight search function for pervasive environments. In *Pervasive*, pp.351–356, 2004.

[2] Stephen DiVerdi, Daniel Nurmi and Tobias Höllerer. A framework for generic inter-application interaction for 3D AR environments, 2003.

[3] Jochen Ehnes and Michitaka Hirose. Finding the perfect projection system – human perception of projection quality depending on distance and projection angle. In *Edwin Sha et al.* , editor, *Lecture Notes in Computer Science*, vol. 4096 , pp. 1017–1026. Springer, August 2006.

[4] Jochen Ehnes and Michitaka Hirose. Projected reality - enhancing projected augmentations by dynamically choosing the best among several projection systems, *IEEE Computer Society*, vol. 0, pp. 283–284, Los Alamitos, CA, USA, 2006.

[5] Jochen Ehnes, Koichi Hirota and Michitaka Hirose. Projected augmentation- augmented reality using rotatable video projectors, In *"ISMAR2004 The Third IEEE and ACM International Symposium on Mixed and Augmented Reality"*, pp. 26–35. *IEEE Computer Society*, 2004.

[6] Jochen Ehnes, Koichi Hirota and Michitaka Hirose. Projected applications-taking applications from the desktop onto real world objects. In *HCII2005 Conference Proceedings [CD Rom]*, *Lawrence Erlbaum Associates*, 2005.

[7] Mototsugu Emori and Hideo Saito. Texture overlay onto deformable surface using geometric transformation. In *ICAT*, 2003.

[8] Matthew Flagg, Jay Summet and M. Rehg. James. Improving the speed of virtual rear projection: A gpu-centric architecture, In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 105, Washington, DC, USA, 2005. *IEEE Computer Society*.

[9] Shinichiro Hirooka and Hideo Saito. Displaying digital documents on real paper surface with arbitrary shape, In *ISMAR*, pp. 278–279. *IEEE Computer Society*, 2003.

[10] Hirokazu Kato and Mark Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system, In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pp. 85. *IEEE Computer Society*, 1999.

[11] Hideki Koike, Yoichi Sato, Yoshinori Kobayashi, Hiroaki Tobita and Motoki Kobayashi. Interactive textbook and interactive venn diagram: natural and intuitive interfaces on augmented desk system, In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 121–128. ACM Press, 2000.

[12] Hanhoon Park and Jong-Il Park. Invisible marker tracking for ar, *ismar*, 00:272–273, 2004.

[13] Claudio S. Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *UbiComp '01: Proceedings of the 3rd International Conference on Ubiquitous Computing*, pp. 315–331, London, UK, 2001. Springer-Verlag.

[14] John Underkoffler and Hiroshi Ishii. Illuminating light: An optical design tool with a luminous-tangible interface, In *CHI*, pp. 542–549, 1998.

[15] John Underkoffler, Brygg Ullmer and Hiroshi Ishii. Emancipated pixels: real-world graphics in the luminous room, In *Alyn Rockwood, editor, Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques*, pp. 385–392. *ACM Press/Addison-Wesley Publishing Co., 1999*.

Jochen Ehnes was born in 1971 in Germany. He received his Masters in Computer Science from the Technical University of Darmstadt in 2000 and his Ph.D. in Advanced Interdisciplinary Studies from the Graduate School of Engineering at the University of Tokyo in 2006. His research interests include augmented and virtual reality, interactive computer graphics, human computer interaction and visualization.

Michitaka Hirose was born in 1954 in Japan. He received BE in Mechanical Engineering, ME and Ph.D. degrees in Mechano-informatics from the University of Tokyo in 1977, 1979 and 1982 respectively. Currently, he is a professor at Graduate School of Information Science and Technology, the University of Tokyo. His research interests include human interface, interactive computer graphics, wearable computer and virtual reality. He was a project leader of Multimedia Virtual Laboratory project and Scalable VR Contents project sponsored by ministry of general affairs. Prof. Hirose is a member of the ACM, JSME(Japanese Society for Mechanical Engineers) and VRSJ(Virtual Reality Society of Japan).