

A Tangible Interface for the AMI Content Linking Device – The Automated Meeting Assistant

Jochen Ehnes

Abstract—In this Paper we describe our approach to support ongoing meetings with an automated meeting assistant. The system based on the AMIDA Content Linking Device aims at providing relevant documents used in previous meetings for the ongoing meeting based on automatic speech recognition. Once the content linking device finds documents linked to a discussion about a similar subject in a previous meeting, it assumes they may be relevant for the current discussion as well. We believe that the way these documents are offered to the meeting participants is equally important as the way they are found. We developed a projection based mixed reality user interface that lets the documents appear on the table tops in front of the meeting participants. They can hand them over to others or bring them onto the shared projection screen easily if they consider them relevant for others as well. Yet, irrelevant documents do not draw too much attention from the discussion. In this paper we describe the concept and implementation of this user interface and provide some preliminary results.

I. INTRODUCTION

While the main purpose of meetings is to facilitate direct communication between participants, documents play an important role in meetings as well. Documents often contain facts that are currently discussed, but they are not necessarily at hand. If these documents were available in a document management system, participants could search for them. However, participants of a meeting usually do not have the time to perform such queries often during a meeting. Therefore a system that could provide relevant documents for an ongoing discussion would be very helpful. A critical part of such a system would be the user interface. It should stay in the background as much as possible in order to not disturb the ongoing discussion by drawing too much attention to it. Yet it should be able to deliver the relevant documents to the participants as directly as possible, so they can incorporate these documents into the discussion directly with minimal effort. In this paper we describe a tangible mixed reality system as an interface for the AMIDA Content Linking Device [1], a system that suggests documents which may be of interest for an ongoing discussion. In the original version the documents suggested by this content linking device were displayed on a laptop screen. Consequently a meeting participant, usually the discussion leader, has to monitor what is going on on the laptop's screen, which certainly distracts him or her from the meeting. Furthermore the laptop's display has the character of a private display. Other participants are not able to see the documents although they may be more important to them. Of course this could

be overcome easily by providing every participant with a laptop showing all the proposed documents. However, then everybody would have to check every document the system suggests. Furthermore, if a participant thinks a document is important, it still would not be straight forward to share it with the other participants. The user would have to describe the document first, so that the others can identify it among all the documents the system suggested so far. All this would lead the attention too much onto the laptop in front of the participants and away from the group. To overcome these challenges we propose to use a user interface projected on the tabletop in front of the participants. By using this less private form of display the documents suggested by the content linking device are visible to other participants as well. By furthermore providing an easy way to grab these documents and move them to other participants' places quickly, they can be moved to the participant they are most valuable to. Documents that are of interest to several participants or that are the subject of the discussion can be moved to a shared space, where they can be looked at by everybody at the same time.

II. PREVIOUS AND RELATED WORK

Started in January 2004, the European AMI (Augmented Multi-party Interaction) integrated project has been building systems to enhance the way meetings are run and documented. AMI research revolves around instrumented meeting rooms which enable the collection, annotation, structuring, and browsing of multimodal meeting recordings. AMIs JFeret browser [2] allows its users to go through previous meetings to get themselves up to date if they were not able to attend these meetings. The browser can display video and audio recordings of all participants as well as the transcript of what was said by whom. Searching for keywords makes it easier to find parts of particular interest. While the possibility to look through recordings of previous meetings and being able to search for important sections by keywords is a very helpful tool, it requires direct action by the user. Furthermore, as the user has to interact with the system on a personal computer, which draws the user's attention to it and distracts from the conversation, the browser is more useful in preparation for a meeting than during the meeting itself.

An important goal of AMI however is to support meetings while they take place. An automated meeting assistant shall find relevant information based on the current discussion and offer it to meeting participants without requiring too much attention from them. The AMIDA Content Linking Device [1] is the first demonstrator of this system. The basic

J.Ehnes is with the Centre for Speech Technology Research (CSTR), School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, UK jehnes@inf.ed.ac.uk

idea is that the assistant monitors the ongoing meeting using automatic speech recognition (ASR). As it gets the transcript from the speech recognition system, it tries to determine the topic of the conversation, searches for relevant documents from previous meetings and displays them.

The idea to use the table top as an interface to computers is not new. The first system of that kind known to the authors was DigitalDesk [3], [4]. Its main intention was to bring together electronic- and paper-documents. In [5], [6] a similar setup consisting of video projector and camera, I/O-Bulb as the authors call it, mounted above the table was used to create applications that are manipulated using physical objects. Applications include the simulation of holography setups using physical placeholders for optical elements such as lasers, mirrors etc. or the simulation of fluids flowing around objects standing on the tabletop. An obvious advantage of this kind of user interface is their collaborative nature as several users can manipulate different physical objects on the tabletop at the same time.

While being able to see what everybody else sees is a very important factor for collaboration, it sometimes is necessary to be able to sketch something down or to look at a document to sort out one's thoughts before presenting them to the whole group. In [7] the authors presented a system that supports the discussion of virtual prototypes by a group of designers/engineers sitting around a projection table. The crucial difference here to other 3D viewers or the applications running on the I/O-Bulbs was that the content does not occupy the whole screen space. Instead the virtual prototype would be visible on a virtual piece of paper, allowing individuals to view them and sketch on them (in 3D) while the rest of the group may be looking at or sketching on something else. While we do not display 3D objects in our content linking system, we use the concept of having virtual pieces of paper that can be moved around using a physical device. The Shared Design Space [8], a system consisting of four projectors for an interactive surface on a tabletop and a projector to create an interactive wall, is of interest as it not only use video cameras to track objects for interaction. Anoto Pens, digital pens that can track a pattern of tiny dots on the surface they are writing on, are used to control the system as well as to draw onto the virtual documents (images). As we aimed for a simple interface to view existing documents however, we do not provide such a feature at the moment.

III. SETUP

In order to present documents found by the content linking device to meeting participants, we planned to build a system that can project these documents on the table tops in front of the participants during meetings. By presenting information of relevance to the current discussion in this way instead of via personal computers at each participants place, we expect the ongoing meeting to be disrupted less than it would if the participants would each interact with their computers. Also, a projection on the tabletop supports the communication within the group as all participants can see the projected documents. Participants should be able to move them around easily, just

as traditional documents on paper. Video tracking seemed a natural fit, as it does not require any additional interaction devices. As one of our goals for the implementation of this system was to provide the additional functionality without requiring big changes to the existing meeting environment, we decided to go for projection and video tracking from above the table. While a back-projection/tracking system from below a semi transparent tabletop would have allowed to detect when objects or fingers touch the surface, having to buy new tables with transparent tops and fitting back projection systems beneath them would have been a too big a change to the existing room as it was already fitted with a lot of recording equipment used by the AMI project. New furniture might have made other changes necessary, which we wanted to avoid to keep recordings comparable. Furthermore, the space required for a back projection/tracking system below the table possibly would become a disadvantage as meeting participants would not have been able to sit at the table as comfortably as usual and projected documents may come and go unnoticed if papers or other objects are put on the desk top on top of them.

To provide enough space for several participants, we planned to use multiple projection/tracking systems. For a start we designed our system to support two users and we used one computer (Mac mini), projector (Optoma EP709) and camera (ImagingSource DFK 21BF04) for each of them. Figure 1 shows our first setup mounted around the projector for the presentation screen. We also included the presentation screen as a shared space into our system, by feeding this projector with a laptop computer (MacBook Pro). Later we replaced one Optoma EP709 projector with an Optoma HD800X HD-projector. More about that in section VII. The software is



Fig. 1. The two projection systems (Mac mini, projector and camera) mounted above the table next to the projector for the projection screen in the old instrumented meeting room. The grey camera is part of the equipment used for meeting recordings.

designed to be scalable, so that we could change the number of projection systems easily. While multiple projectors are often used to create a larger, tiled display [9], [10], that approach did not appear to be so suitable for our application.

Desks in meeting rooms often are not arranged to create one large surface, but in different shapes, such as a "U" shape, to allow everybody to see everyone else as well as a projection screen. Some tables may not even be connected to others at all. In consequence it is not important that the projection systems form a continuous display area, as long as the user interface is consistent across the systems and documents can be moved between them in a way that is consistent to the way they are moved around on one unit. Furthermore, this approach gave us the flexibility to provide individual display modes for each user, such as the shared display layer described in section IV-C.

IV. USER INTERFACE

In order to make the interaction with the system as direct as possible, we aimed to make the projected objects graspable. We decided against hand tracking as it is difficult to distinguish between gestures meant to manipulate documents and gesturing during discussions. This is especially true as the current setup does not allow to detect if the user's hands touch the desk top. Instead we track physical objects that serve as interaction devices using the AR-ToolKit+ tracking library.

A. Document Handling

Documents can be moved around using a physical object (paper grabber). By placing this grabbing device onto the projected document and covering the switch marker, the document gets connected to the device. As long as this connection exists, the virtual document follows the grabber. And it does so not only on one users space, but across multiple projection units. Furthermore, this connection continues after long pauses of visibility, even between different runs of the projection system. The grabber objects consist of a piece of cardboard containing three markers, one of them elevated on a box (see figure 2).

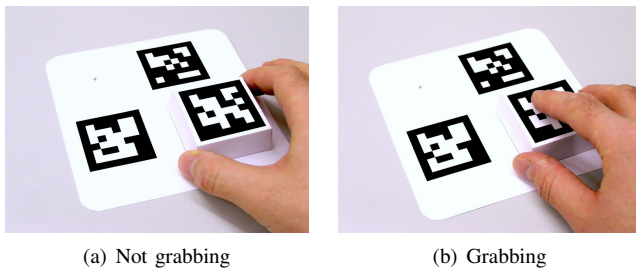


Fig. 2. The grabbing device and its switch marker.

This box can be grabbed easily to move the device around. If grabbed only on the sides, so that the marker on top is fully visible, the grabber is disconnected and can be moved freely. Once the grabber is placed on a virtual piece of paper, users can grab the document by holding the box like a mouse and thereby covering the top marker. Once grabbed, the document stays connected to the grabber until it is released again, i.e. the top marker gets recognized again. This may be on the same projection system or that of another user.

B. Keyboard Forwarding

Instead of providing virtual, projected keyboards as it is usually done with touch screen interfaces, we chose to use standard wireless keyboards. In order to allow

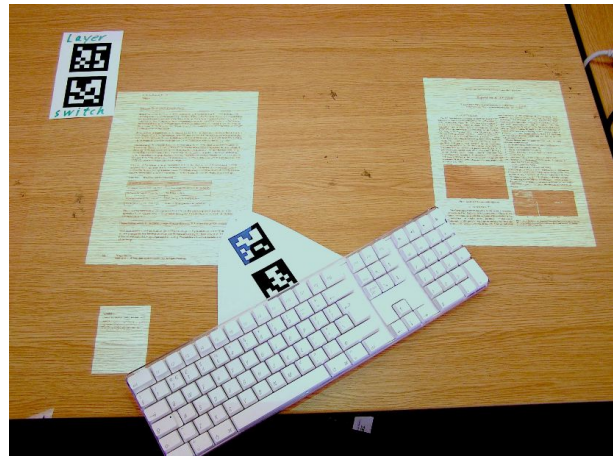


Fig. 3. Wireless keyboard getting connected to a document by placing its marker on the document.

keyboard based input, a Keyboard identified by the two markers attached to it (see figure 3), can be placed on a displayed document. This allows to route keyboard events to the display applications (section V-C) that create the graphical representation of the documents. It replaces the physical connection (which keyboard is connected to which projection unit) with a virtual connection between keyboards and documents. Once a keyboard's markers are placed over a document, the document gets connected to the keyboard and receives events of pressed keys. Hereby it does not matter if the keyboard is connected to the projection unit that displays the document or another one. The connection stays active until the keyboard is connected to another application. In the PDFReader application the arrow keys allow users to change the displayed page of a multi page document.

C. Sharing

In addition to augmenting the table, we wanted the system to incorporate the whiteboard as well. This way, participants are able to interact with content on the whiteboard directly from their place and move content between their space and the shared whiteboard space easily. While Hyperdragging as described in [11] would allow participants to do that in principle, it relies on a laptop with a conventional interaction device such as a touchpad. Using hyperdragging therefore would work against our goal to let the computer disappear.

We believe it is better to 'bring the shared screen to the participant' on the press of a button, or in our case when a marker is covered by the user. We therefore implemented a shared semitransparent layer (see figure 4) on top of the normal projection area which can be activated and deactivated by covering a marker placed on the projection area for that purpose. The presentation screen is the only exception here, as it does not have a private layer. It always displays the

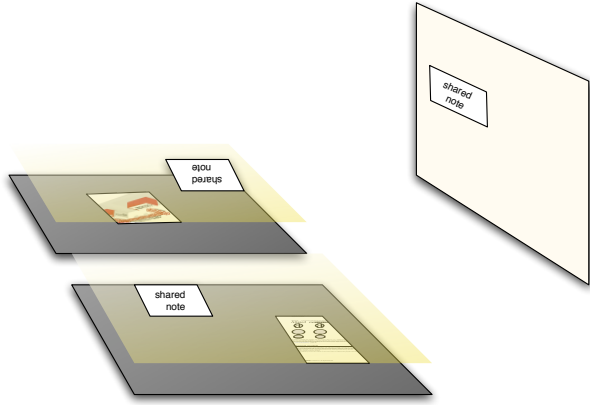


Fig. 4. The shared space as a semitransparent layer on top of the normal space. It can be (de)activated by covering a switch marker.

shared layer. Documents can be moved between the private and shared layers by grabbing them on one layer before switching to the other one. Once on the shared layer, all state changes such as position, orientation or which page of a multi page document is shown are forwarded immediately to other systems displaying the shared layer.

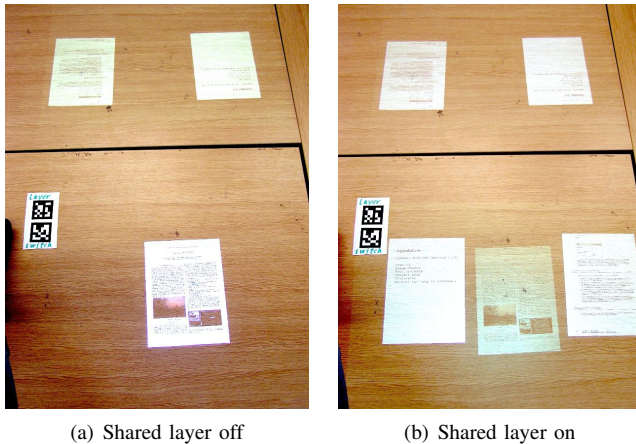


Fig. 5. Switching the semitransparent shared layer (lower part, upper half is the shared layer on the 2nd user's space).

D. Auto-Arrangement and Auto-Iconizing

The Content Linking Device brings up new documents in regular intervals. In fact it often finds several documents to be displayed at the same time. In this situation it is obviously not enough to make the documents appear at a fixed location such as the center of the tabletop. Of course the space on the table is not unlimited, so a method had to be developed to prioritize documents and remove less relevant documents gradually. We implemented a system to arrange and iconize documents automatically. It behaves as follows: When a display application is started to present a document, it is appended to the array of automatically arranged applications. If the number of elements in this array is growing above a given limit (two applications in our case), the first element is

removed and appended to the array of iconized applications. Additionally, a timer is set for each application added to the array of automatically arranged applications. Once the timer fires, the document gets iconized as well. This way documents that do not appear to be relevant to users are removed as well. If the number of elements in the array of iconized applications grows above its limit (ten applications in our case), the first application is removed and terminated. Whenever applications are added to or removed from these arrays, the applications are sent a new goal position and scale factor according to the array and their position within that array. The first auto arranged application is displayed on the left side. The second (and latest) one is positioned next to it in the center of the projection, leaving the right side for documents the user places there to read. Their scale factor is 1.0 so they are displayed in full size. The automatically iconized applications on the other hand are scaled down to 0.3 and arranged along the front edge of the table with the oldest one being displayed on the left and new ones being added to the right. When applications are sent to new positions or receive new scale factors, they don't change to these values instantaneously. Instead they animate towards these values over a given duration (1.5 seconds seemed best). This way it becomes obvious when the layout changes and it is easy to follow what is going on. This is very useful when an application that is already open is deemed to be relevant by the content linking system again, as one can see the document move from its previous position to the position of the newest document (center). If the user places a paper grabber or keyboard connector on top of a virtual paper, it prevents the paper from being affected by the auto arrangement/iconizing system. If placed on an iconized paper, the paper is also scaled back up to full size. Now the user may move the document to where it can be read conveniently without interference of the auto arrangement system. Once the user removes the paper handling device and/or keyboard connector from the projected document, the system will again take responsibility for it and iconize it after 30 seconds to clean up the tabletop. For the case that the document the content linking device determined as relevant is already displayed as the latest document or controlled by the user, a visual ping has been implemented. If pinged, a document visually vibrates for a short period of time. It is animated by scaling it slightly up and down from its original size using a sine function. The amplitude of this vibration is scaled down to zero within 1.5 seconds to fade out the effect smoothly.

V. SYSTEM ARCHITECTURE

In order to keep the number of projection systems scalable, we divided our projection system into two parts. Smart Projector, an application running on all projection units and a central Projection Manager. This display system is connected to the Content Linking Device via a third application, the Hub Manager (figure 6). These applications may all run on different machines as well as the Hub and other components of the content linking device ([1]). In our test setup however,

we ran all these components on the laptop that rendered the contents of the shared screen. Only the two instances of Smart Projector augmenting the users' places, which had to perform video tracking as well, ran on separate machines, the two Mac minis.

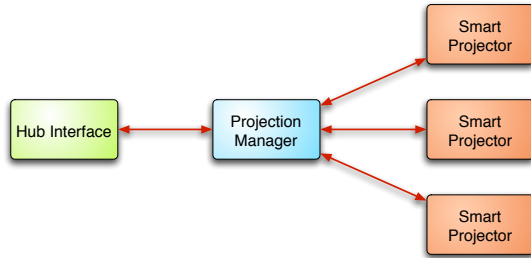


Fig. 6. System architecture.

A. Smart Projector

Smart Projector is the application running on every projection unit that creates the actual user interface for meeting participants. While it has a simple user interface to connect to / disconnect from the Projection Manager and to configure and activate the capturing of life video from the camera, it is switched to Fullscreen mode during normal operation. If a video input stream is available, the application searches for AR-ToolKit+ markers in it and sends the information to the Projection Manager and the applications associated with the tracked object. If no video stream is available, it displays shared documents (section IV-C) only. Additionally it also captures key board events and forwards them accordingly (section IV-B).

B. Projection Manager

At the center of this projection system is Projection Manager, a server that manages all important parameters of the projection units and coordinates their actions. As the Mac minis only have one display connector feeding the projector, most parameters of Smart Projector (background color, projection parameters, calibration between projection and camera coordinates, ...) are adjusted in the Projection Manager. This also makes it easy to adjust parameters for several projection units at once.

Besides managing the projection units, Projection Manager is also used to define and print the interaction devices carrying AR-ToolKit+ markers (see figure 2 for example) and to manage the display applications running on the system.

C. Display (Hydra-) Applications

In order to keep the system easily extensible to new types of documents, we developed an API we called *Hydra Applications*. In a way analogous to the creature from ancient greek mythology which had one body and could grow several heads, Hydra Applications run on one projection unit (the body and first head) but can display their content on several

projection units (additional heads). In order to facilitate that the API consists of two base classes, one for application objects, the body in our analogy, and one for display objects, the heads. The display objects are basically stateless objects that render the content they are sent by the application object on the table top in the form of a virtual sheet of paper. They are also responsible to forward certain events, such as changes to position and orientation of tracked devices or keyboard events, to the application object. The application objects on the other hand are responsible to maintain the state and change it according to user input. Whenever relevant parameters change, the application objects send updates to their display objects. This separation of state and display allows for an easy duplication of the display part without creating inconsistencies. When a display application is started on one system while it is not running anywhere else, a display object and an application object are created and connected to each other on the same projection unit. On the other hand, if the application is already running somewhere else, only a display object is instantiated. This display object is then connected to the application object on the projection unit where the application was started first. After that the application object receives a call to update its display objects, so that the newly created display object gets all the information needed to display the correct data. This happens in our system when a projected piece of paper gets moved to the shared space (section IV-C). In order to be able to display the document on all projection units' shared spaces, its display application is started on them as well. Now events from the display object on any projection unit (eg. grabbing and moving the document, or keyboard events) are forwarded to the application object, which in turn updates its state and sends updates to its display objects on all projection units. The only application we currently use with the content linking device is a PDF viewer. As this is mainly a display task, the display object contains most of the functionality here, which is to render the current page of the PDF file to a texture and display that texture on the tabletop at the correct location and orientation. The application object only loads the PDF data, sends to the display objects and maintains a minimal state, which consists of the number of the current page. It increases/decreases the number according to keyboard events and sends the new value to the display objects after a change. It also sends it to the Projection Manager before it terminates and gets it from there when it starts up the next time to keep the experience consistent between different runs. This happens for example when a document is grabbed and then the grabber is removed from the tabletop. The application terminates and the document disappears. Later, when the grabber is recognized by a projection unit, the application is started up again and continues from where it left of before it was terminated. Another Hydra Application we developed may be compared to a post-it note. It displays simple text that can be edited. Here the string that makes up the message represents the state managed by the application object. It is sent to the display objects after any changes and stored on the Projection Manager between different runs. While

this application could be useful for brainstorming, we are currently not using it with our automated meeting assistant system.

D. Hub Interface

To connect the display system to the AMI Content Linking Device the Application Hub-Interface was developed. On one side it connects to the Hub as a consumer via the Java Native Interface, and on the other side it connects to the Projection Manager. Once the query aggregator stores new related documents in the Hub (please refer to [1] for a detailed description of the components of the content linking device), the application receives a message from the hub containing the documents name. As the documents themselves are not stored in the Hub, they have to be loaded from a different location. For this prototype we converted all documents to PDF and placed them in one directory. The path to this directory can be specified in the Hub Interface's GUI. For testing purposes the Hub Interface can display these PDF documents in separate windows after receiving the message from the hub. However, the standard functionality is to introduce these documents into the projection system. It does so by adding a new PDFReader application to the list of applications and setting the Document URL as well as the Application ID to the URL of the PDF file. Then the application is started on a projection unit. Which projection unit it is started on is currently specified by a popup menu. In a future version the content linking system should also provide a person or role for whom the document is most relevant. Then the documents can be sent to the best fitting person automatically. If the document is already being displayed, it is not introduced a second time. Instead the user is notified that the document could be relevant for the current discussion by bringing it to the front and letting it vibrate a little bit to create a visual ping (section IV-D).

VI. FIRST RESULTS

A weakness of our first system was the resolution of its projectors (1024x768). While this is ok for documents containing little text in a large font, such a meeting agenda or Power Point slides, it clearly was not sufficient to display regular text documents as a virtual sheet of A4 paper. Figure 7 shows a PDF document projected onto a paper version of it. While projected headings are readable, the projected text clearly is not.

In order to make the system really useful, it needed a much higher resolution. Mark Ashdown [12] proposed to use two projectors, one to cover the whole table top with a relatively low resolution and one to cover a small area with a high resolution. However, we believe that this would effectively limit the usable display area to the small area of high resolution projection and as such would have a negative effect on our system. Participants would have to move all documents onto this foveal display area to be able to read them, which would introduce additional load to the user. It would no longer be possible to just throw a glims at a new document. Furthermore, it would also make it impossible

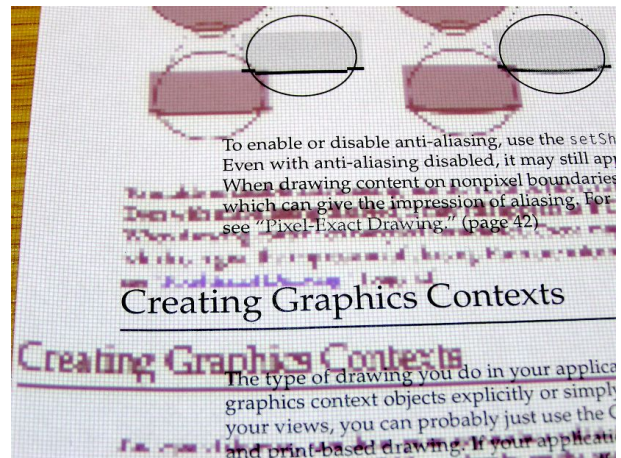


Fig. 7. Electronic document projected onto a printed version. The small text is unreadable at the resolution offered by our first prototype.

to place a virtual paper between two participants to have a look at it together, as that would be in the low resolution area. Consequently we believed that it would be necessary to use projectors with a higher resolution or to tile multiple projectors to achieve a high resolution everywhere.

VII. IMPROVED PROTOTYPE

In order to address the resolution problem we used a projector with a higher resolution. In our new instrumented meeting room we replaced one of our Optoma EP709 projectors with an Optoma HD800X, a projector capable of projecting full HD video (1920x1080). In order to get an even better resolution from this projector, we used it in portrait mode, effectively augmenting only half of the users' table space. As documents are usually printed in portrait format, this enabled us to make better use of the projected pixels. While the EP709 projector would give us 768 pixels in height, the HD800X in portrait mode gives us 1920 pixels for the same height, an increase in resolution by a factor of 2.5 in one direction. Figure 8 shows the projection areas for comparison, the larger area created by the lower resolution projector on the opposite side of the table and the smaller area originating from the HD projector on the front side. Figure 9 gives an impression of the quality of a paper projected onto the tabletop by the HD projector. The text is now readable. For comparison the document can be found at <http://www.ijvr.org/reports%20and%20experts/report.icat.pdf>.

While normal sized text can now be read on projected papers, the fact that only about half of the area on the table in front of a user can be used as display surface creates another limitation. We have to investigate if the resulting area is sufficient to present all the documents arriving from the content linking device.

Similarly we have to test if the fact that only half of a user's space can be used to present documents is a serious limitation, or if they would prefer to use other objects, such as paper documents or their own laptop besides the projected documents anyway. Also, in order to facilitate collaboration

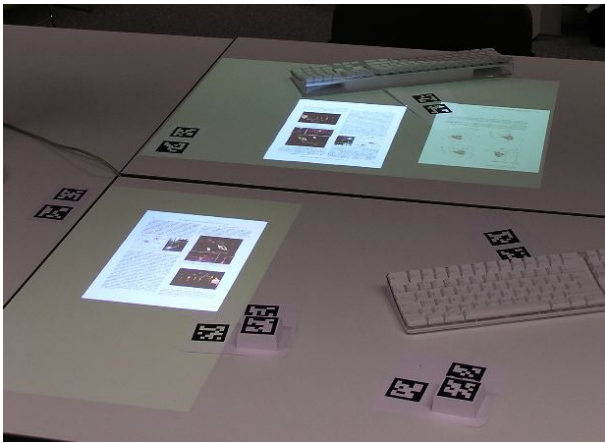


Fig. 8. The high resolution projection in the front left. Both shared layers have been activated to show projection area.

it may be sufficient to place the projection area at one side of users' work areas, so that it can be shared with the neighbour on that side. Then one could still move the document to the other neighbour's space to share it with him/her. Of course, if that should not be sufficient, we could still install two HD projectors for each user. However, that could lead to serious thermal issues in an ordinary meeting room, especially if we want to increase the number of participants to four (all meetings recorded for AMI so far had four participants) as well.

VIII. CONCLUSION

We developed a scalable projection system to be used in meeting environments. The way it is set up allows for easy installation in existing environments. After all, the camera, projector and computer can be integrated into a single unit mounted above the tables. We implemented software components that allow for easy management and coordination of the projection units as well as a user interface based on tracked interaction devices. We demonstrated that it is easily possible to move documents around on one, as well as between different projection units or between private and shared spaces. Figure 10 shows the system projecting documents onto the table top and the room's projection screen. With our second prototype in the new instrumented meeting room, we also demonstrated that even documents with normal sized text projected by our system are readable. Furthermore, the system is able to connect to the central Hub of the AMI project. This way it can be used to display documents the content linking device deems relevant for the ongoing discussion. Additional functionality to manage the displayed content automatically was implemented to cope with the stream of new documents being introduced by the content linking device.

As it turned out though, the first version of the content linking device we used in our prototype had not been tuned

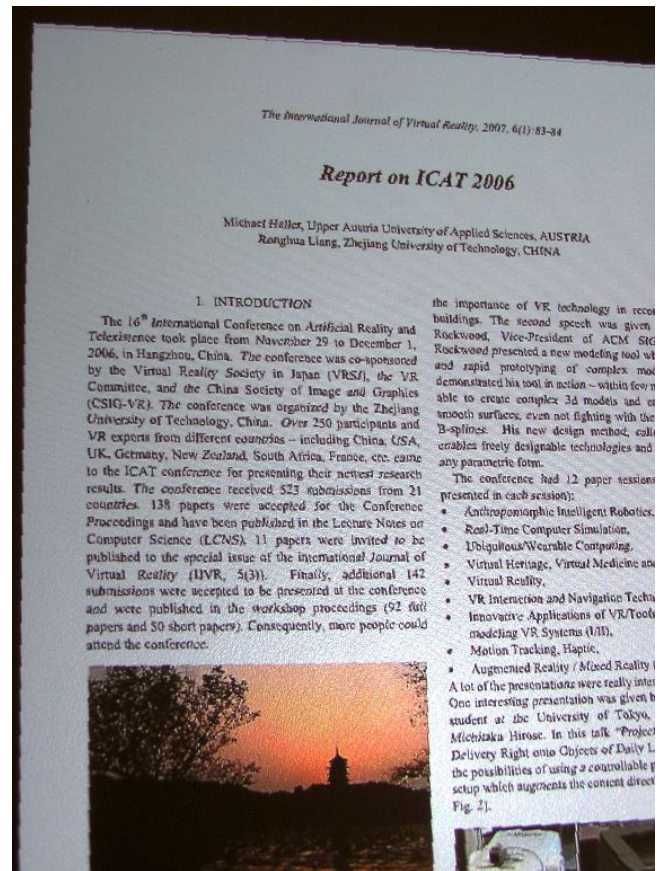


Fig. 9. Papers are readable with the HD projector augmenting half of the table.

well enough yet. During the trial meeting¹ it repeatedly brought up two agendas but rarely anything specific. Also, the system currently does not take into account the role of different meeting participants. During the meetings that were recorded as part of the AMI project, each participant has a particular role such as project manager, marketing specialist, designer etc. and information about this is stored in the recordings as well. We believe that it would be beneficial if the content linking device could assign the documents it wants to introduce to specific roles/participants. This way the document could be introduced directly in front of the person it is most likely to be useful to. Currently, without this functionality, all documents are introduced on one projection unit. This leaves the person sitting at this place (or the immediate neighbours) to decide for whom a document might be relevant and to hand it over to that person.

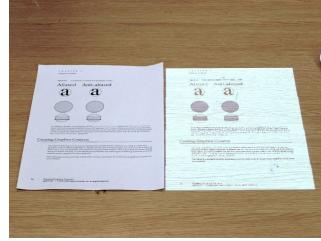
IX. FUTURE WORKS

Our next step will be to update the version of the content linking device we are using. We hope that this will increase the usefulness of the documents found. Furthermore, the new

¹Meeting ES2008d of the AMI Meeting Corpus [13]. For testing purposes meeting ES2008d is played back as input to the Content Linking Device, which searches for relevant documents from the meetings ES2008a/b/c. For more details please refer to [1]



(a) Shared documents at the users' places and the screen



(b) Printed and projected document side by side

Fig. 10. Projection system augmenting two users' tabletops and the room's projection screen.

system also searches Wikipedia entries that could be useful and our system would have to include those as well.

Once the content linking device gets better tuned, we plan to include the system in the scenarios for future AMI meeting recordings. This should give us the possibility to evaluate the system in a formal way. In that context we also will have to find out if one HD projector per user (augmenting half of the user's desk top area) is enough, or if we have to use two of them instead.

X. ACKNOWLEDGEMENTS

This work is supported by the European IST Programme Project FP6-0033812 (AMIDA) as well as the Marie Curie Intra European Fellowship (IEF) FP7-221125 (NIPUI). This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

REFERENCES

[1] Andrei Popescu-Belis, Erik Boertjes, Jonathan Kilgour, Peter Poller, Sandro Castronovo, Theresa Wilson, Alejandro Jaimes, and Jean Carletta. The amida automatic content linking device: Just-in-time document retrieval in meetings. In Rainer Stiefelhagen Andrei Popescu-Belis, editor, *Machine Learning for Multimodal Interaction*, number 5237 in Lecture Notes in Computer Science, pages 272–283. TNO, Springer, 2008.

[2] Michal Fapso, Petr Schwarz, Igor Szöke, Pavel Smrz, Milan Schwarz, Jan Cernocký, Martin Karafiát, and Lukás Burget. Search engine for information retrieval from speech records. In *Proceedings of the Third International Seminar on Computer Treatment of Slavic and East European Languages*, pages 100–101, 2006.

[3] Pierre Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. In *Proc. ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 107–115., 1991.

[4] Pierre Wellner. Interacting with paper on the DigitalDesk. *Communications of the ACM*, 36(7):86–97, 1993.

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

[6] 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*, pages 385–392. ACM Press/Addison-Wesley Publishing Co., 1999.

[7] Jochen Ehnes, Christian Knöpfle, and Matthias Unbescheiden. The pen and paper paradigm supporting multiple users on the virtual table. In *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, page 157. IEEE Computer Society, 2001.

[8] Michael Haller, Peter Brandl, Daniel Leithinger, Jakob Leitner, Thomas Seifried, and Mark Billingham. Shared design space: Sketching ideas using digital pens and a large augmented tabletop setup. In Zhigeng Pan, Adrian David Cheok, Michael Haller, Rynson W. H. Lau, Hideo Saito, and Ronghua Liang, editors, *ICAT*, volume 4282 of *Lecture Notes in Computer Science*, pages 185–196. Springer, 2006.

[9] R. Raskar, G. Welch, and H. Fuchs. Seamless projection overlaps using image warping and intensity blending. In *Proceedings of the Fourth International Conference on Virtual Systems and Multimedia*, Gifu, Japan, November 1998.

[10] heyewall. Heyewall. <http://www.heyewall.de/>.

[11] Jun Rekimoto and Masanori Saitoh. Augmented surfaces: A spatially continuous workspace for hybrid computing environments. In *Proceedings of CHI'99*, 1999.

[12] Mark Ashdown and Peter Robinson. A personal projected display. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 932–933, New York, NY, USA, 2004. ACM.

[13] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: A pre-announcement. In Steve Renals and Samy Bengio, editors, *MLMI*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2005.