A Wall-sized Focus plus Context Display

Sebastian Boring, Otmar Hilliges, Andreas Butz
University of Munich, Media Informatics
Amalienstrasse 17, 80333 Munich, Germany
{sebastian.boring, otmar.hilliges, andreas.butz}@ifi.lmu.de

ABSTRACT

In this paper we present a wall-sized focus plus context display, consisting of three back-projected displays and a steerable projector for output. For input tracking we use a commercial SmartBoard with high precision optical tracking and four additional webcams which provide lower precision tracking on the entire wall. Our approach hence offers two different resolutions in both the output and input direction and thus implements a two-way focus plus context concept with three levels of interactivity. We describe the implementation of such a display and discuss the tracking issues encountered. In order to test the display, we have implemented two example applications to explore its capabilities.

1 INTRODUCTION

With the increasing availability of large display technologies, such as powerful projectors or large active display panels, these are getting more and more attention in the research community, either as public information displays [1] or as shared interaction spaces [17]. When considering large interaction spaces, these are normally assumed to be uniform in terms of resolution and interactivity. An exception is the class of focus plus context displays introduced by Baudisch et al. [2], which is non-uniform regarding their output resolution. This paper presents a wall-sized display which is also non-uniform regarding input.

Wall-sized displays bear great potential for co-located collaboration since several users can work with them simultaneously. When interacting closely, each user can only comfortably overlook a certain section of the display. By stepping back and forth, users can adjust the size of this section and gain more and more overview, but loose the ability to directly interact. A variable concept of focus plus context is thus inherent in dealing with large displays. When users are close enough to interact, certain parts of the wall are also out of their physical reach. This suggests that fine-grained interaction in the outer areas is not needed in this situation.

Large displays can contain huge quantities of data, but humans can only process limited amounts of data at a time. Human perception reacts to information superabundance with the mechanism of guided visual attention, where only a single object or restricted area in our field of view is processed in full detail and everything else is only virtually represented in our memory (e.g. the cup we try to grasp is in the focus, the surroundings are not visually processed in full detail). Tasks in which several independent objects need to be coordinated (e.g., juggling) are handled by time-sharing of attention or, in other words, rapidly switching the focus between objects [20].

Baudisch et al. [3] present a focus plus context display to support the perceptual phenomenon of focus on a technical level. It contains a detailed display in a central focus area, while providing coarser displays in the outer context area. This scheme is very adequate for human perception while at the same time preserving technical resources, such as rendering power.

While Baudisch's focus plus context display is uniform regarding input, because it simply uses mouse input, we present a wall-sized focus plus context display, which makes different parts of the wall interactive to different degrees. We argue that this systematic extension of the original concept can easily be applied to large interactive surfaces, such as wall displays.

2 RELATED WORK ON FOCUS PLUS CONTEXT AND LARGE DISPLAYS

Baudisch et al. [3] describe the principle of focus plus context displays and show how they allow users to manipulate detailed information while keeping an overview of the complete data. In their work they consider focus plus context displays for single users with a mouse and a keyboard as input devices.

Large and wall-sized displays have been used in instrumented rooms, such as *iWork* [11] or *Roomware* [17] with the *DynaWall* [8]. They mostly use off-the-shelf SmartBoards to allow interaction with the displayed information. Thus, this work does not provide any interaction possibilities on a concrete wall with front-projected information. In addition, Rekimoto et al. describe the concept of *continuous workspaces* [19] which allow users to treat different display surfaces as one continuous screen.

While some current work on large displays concentrates on interaction techniques, which just transfer the desktop metaphor to a bigger size [4][6][21], others try to devise novel interaction concepts which are better suited to the particular interaction situation [5][14]. One of our demo applications, Brainstorm, is strongly influenced by Igarashi's work on sketch-based interfaces [10] and his earlier work on Flatland [15].

3 DESIGN OF THE WALL-SIZED FOCUS PLUS CONTEXT DISPLAY

Display Setup

Our main goal was to make a wall entirely interactive, so that it would appear as one big logical screen to users as well as programmers. At the same time, it should support different degrees of interactivity in different areas. In order to provide high resolution output at regular working heights, we embedded three back projection displays of 147 x 112cm into the wall which has overall dimensions of 4.50 x 2.40 meters. The resolution of these three screens together is 3072 x 768 pixels, which corresponds to a spatial resolution of 0.8 pixel per millimeter.

The rest of the wall is covered by a steerable projector – the Beamover 40 from publitec [18] – mounted in the ceiling, which creates a movable display area of about 60x45cm at a spatial resolution of about 0.3 pixel per millimeter. The projected image is rectified using a technique similar to the ones described in [16]. This makes the entire wall above and below the screens a time-multiplexed low resolution display.

General Input Setup

Users should be able to interact on the wall with their bare fingers. In the center of the wall, i.e. on the middle screen, we wanted the input to be very exact and fast, in order to support detailed interaction in a focus area. This was achieved by a SmartTech SmartBoard [22]. The surrounding areas of the wall as well as the side displays, serving as context areas, were meant to have a lower input resolution. We decided to use optical tracking for this, as it would allow bare finger interaction.

As the system is wall-sized, multiple users can stand in front of the display and interact simultaneously. Hence the system needs to support simultaneous input. The wall should appear as one interactive display to the user as well as the programmer. Thus none of them should be aware of the tracking technology currently used. This requires seamless transitions between the technologies while users interact with the wall display. The combination of different tracking systems, together with the simultaneous multi user input, bears several technical challenges.

Hardware Setup for Camera-based Tracking

The setup of the wall-sized tracking system mimics the SmartTech *DViT* technology [22] by arranging four cameras in the four corners of the wall. In order to cover a large part of the wall, we used Logitech's QuickCam Fusion [13] with a resolution of 640 x 480 pixels, 30 frames per second and a diagonal field of view of 72 degrees. Other cameras had either a lower resolution or a lower field of view. To ensure a fast detection close to real-time processing, the tracking system runs on an Intel Pentium 4 workstation with 3.0 GHz with Hyper Threading including 1 GB of RAM. This computer is dedicated to the camera-based tracking system.

Because the webcams have a field of view of 72 degrees we needed to decide how they will be arranged on the wall to match the following requirements: First, we wanted to have at least two cameras viewing every sub-area of the wall. Second, most of the wall should be covered by the cameras. Third, the coverage area of three or four cameras should be maximized. Out of several possible arrangements (see Figure 1) we chose to use setting (*c*).

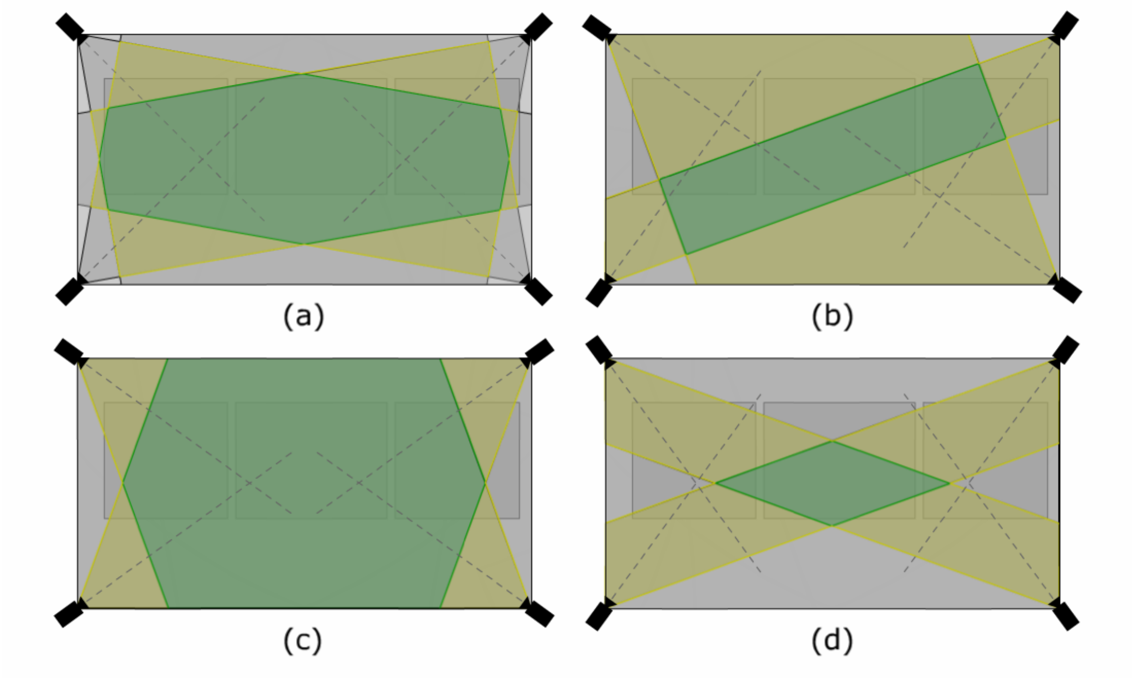

Figure 1: (a) shows the angle of aperture of each camera centered (at 45°), (b) visualizes the cameras aligned to one side of the wall (every camera to a different side), (c) shows the four cameras aligned to the bottom and top side of the wall respectively and (d) represents all cameras aligned to the left and right sides of the wall respectively. Green areas represent the coverage of all four cameras while yellow areas show the coverage of three cameras.

The cameras needed to overlook the entire wall with their optical axis parallel and close to the wall's surface, but space did not allow t to be embedded into the wall. Therefore we used a mirror construction with cameras facing directly towards the wall. This allows observing a small rectangular area over the wall's surface. First, we constructed the ideal fixations in a 3D rendering program that allowed us to test our formulas and parameters. Figure 3 shows design drawing of the desired fixations. These have been built from wood and hold the mirrors. In addition, we attached black foamed rubber to avoid that the cameras capture too much light (see Figure 2).

Figure 2: Different views of the fixation for one camera. Left and center show the fixation without the black foamed rubber on it. Right shows the final fixation attached to the wall.

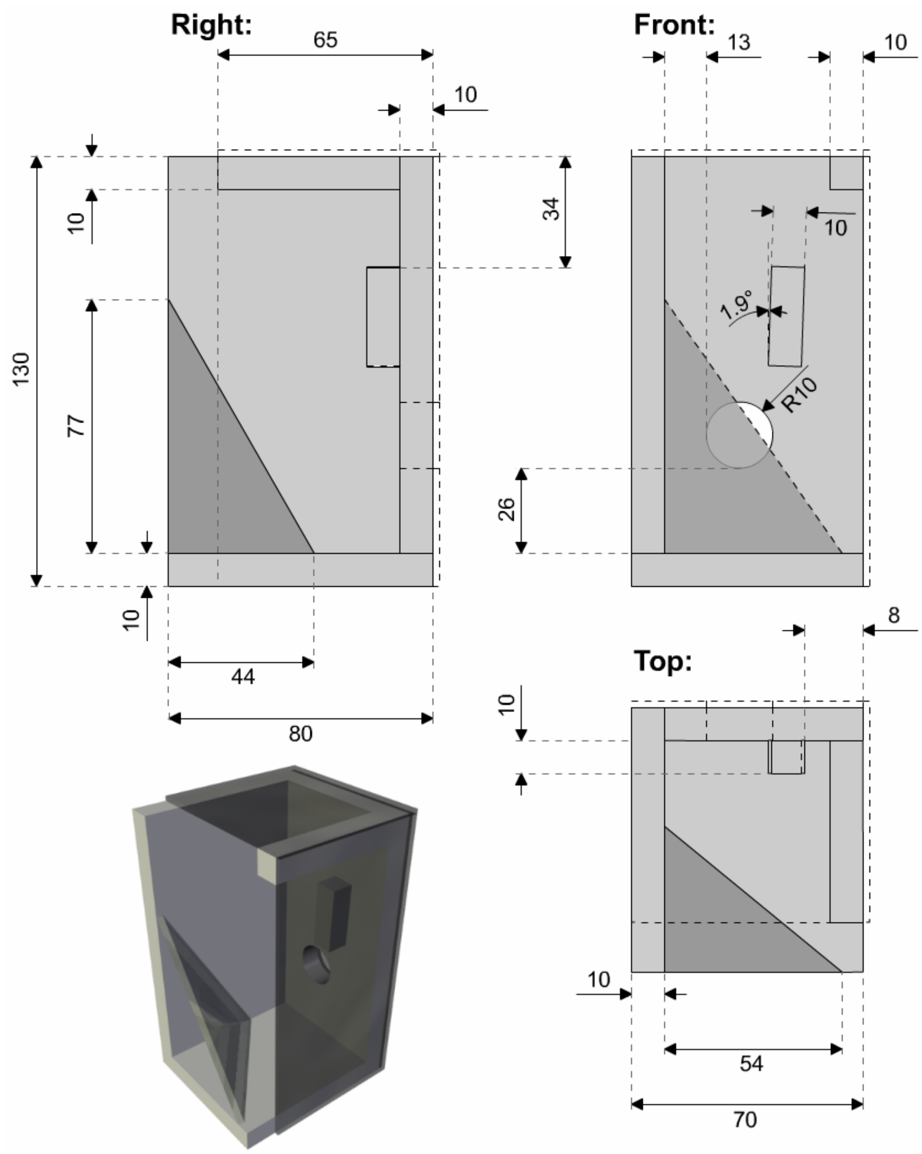

Figure 3: Design drawing of the fixation for one camera. Unit of measurement is millimeter.

4 RELATED WORK ON FINGER TRACKING

To make displayed information interactive, multiple tracking technologies have been developed. Most of them use infrared light for increased robustness and accuracy, such as *DViT* [22] or *HoloWall* [14]. As *HoloWall* has back-projected infrared light, interaction on a solid wall is impossible. Another approach is to use *total frustrated internal reflection,* which takes advantage of the fact that light is refracted to a certain extent if traveling between two different media [9]. Again, the surface needs to be seen by a camera from behind, which is impossible on a solid wall.

A further technique is to use capacitive sensing as it is implemented in today's touch screens and touch pads. Paul Dietz et al. [7] describe one example called the *Diamond Touch*. Their users act as an active part (coupling devices) of the system. The antennas are implemented as arrays in the interactive displays whereas the receivers are attached to the chairs the users sit on. Building antennas into our back projected displays would disturb the projected image.

5 REAL-TIME FINGER TRACKING

In order to achieve a reasonable precision of about 10-20mm with our relatively low resolution cameras, we chose angulation as the base tracking technology: Several cameras determine the position of a finger in the image and from this derive the angle between their optical axis and the connection line between camera and finger. With two cameras at known positions and orientations, the intersection of these lines is the finger position. In order to track more than one finger, additional cameras are needed to resolve ambiguities. Our four camera setup can track four fingers at a time. In order to increase robustness and reliability, we introduced several modifications and additions to the base method.

Calculating Angles from Live-Captured Images

To calculate the final positions of fingers on the wall it is necessary to gather the angles from each finger to the four cameras' live images. These images need to be analyzed by the system to get the relevant information (e.g. the fingers) within them.

It is important that the system will not detect non-existent objects (*false positives*) or let existent objects be undetected (*false negatives*). As described earlier, the system has four webcams mounted in the corners of the wall that observe a small stripe over the wall's surface. As our room has white walls we initially assumed that we could use simple frame differencing – current frame minus calibrated frame. But as soon as the displays are turned on, their radiation changed the color of the user's finger as well as the color of adjacent walls into the display's color. Thus, we needed to find a unified background that will not change its color due to display radiation. We finally decided to use black velvet mounted onto the adjacent walls (see Figure 4).

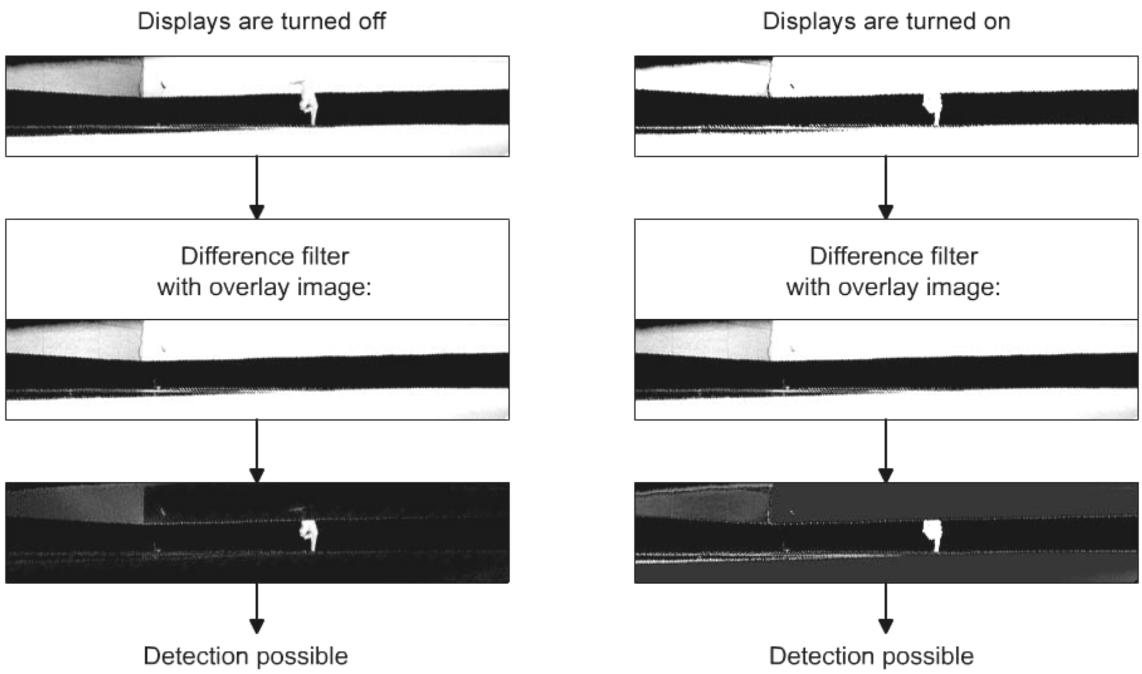

Figure 4: Left shows the detection with black velvet mounted on the wall with no display radiation. Right shows the same setting with bright displays resulting in radiation.

By detecting stripes in the captured images we can determine the respective angle. The calculated angle β will be the angle between the camera's normal and a virtual line from the camera's origin to the detected point. This can be done knowing both the camera's angle of aperture α and the width w of the captured image using equation 1.

$$\beta = \arctan\left(\frac{2 \cdot \Delta x \cdot \tan \alpha/2}{w}\right), \Delta x \in \left[-\frac{w}{2}; \frac{w}{2}\right]; \text{ (Equation 1)}$$

Every horizontal pixel value within the image corresponds to a different angle. The width *w* of the captured image in pixels determines the tracking resolution and is therefore a significant identification of the system's tracking quality.

Detecting multiple Points

Once we have the calculated angle(s) of all cameras, the system can start to calculate the positions on the wall. We will first describe how one intersection can be found. The idea of triangulation is to create two virtual lines *g* and *h* from each of the cameras *A* and *B* with the known angles α and β. The intersection of these virtual lines can be calculated with simple vector mathematics and results in the finger position *P*.

If a system has two cameras and each camera detects two points (i.e., two angles) the system would have to intersect each line of camera *A* with each line of camera *B*. This results in four intersections – and hence in four detected positions - although there are only two fingers. Thus, angulation cannot be used in its original form for more than one finger because this will lead to *false positives*. In addition, having more than two cameras increases the system's robustness and accuracy.

Due to the limited resolution of the webcams, a calculated intersection will never be the exact intersection of four lines. To address this problem, the system needs to allow points to be close (e.g. within a given threshold) to another line. However, four cameras can only track multiple fingers correctly, if no finger is occluded by another finger. To resolve this, we defined further criteria regarding a detected intersection. The set of restrictions an intersection must fulfill to be an accepted position is:

- **Clearness of involved lines:** An intersection must provide at least three lines without having a pair of them nearly parallel. In particular, the angle between involved lines must not be less than a certain threshold. Otherwise, the two lines will become one involved line regarding this intersection.
- **Unambiguous mapping of lines to intersections:** Two intersections must not have the same lines involved: at least three lines must be different. If this happens the more accurate intersection will be taken while ignoring the second one. Due to occlusion problems it is important to have at most one line involved in multiple intersections.

Given the second criterion it is obvious that this tracking method will not work in the center region of the observed surface. This is nicely compensated by the overall setup of our focus plus context display with the high resolution input area in the center.

Continuity of Finger Movement

Once the system has detected the positions on the wall it still does not know which finger caused the position information. In the case that the system is restricted to detect "*finger up*" and "*finger down*" events this information would not be necessary. In our case, it also needs to be aware of the movement of fingers. In particular, the system needs a clear association between currently measured and previously detected fingers.

We implemented a simple continuity method called "*Dead Reckoning*" (Deduced Reckoning) [12]. In general this is another positioning method assuming that – in addition to the last known position – either the velocity or the distance to the next position is known. As the system does not know both of the parameters, it assumes that the orientation and speed of the finger will only slightly change until the next position has been detected.

Whenever a position is recognized by the system for the first time, the orientation (vector) will be set to zero. As soon as the system associates a newly detected position P_{i+1} with a previously recognized finger position P_i, it replaces the orientation v_i with a new one (v_{i+1}). To associate a newly detected position with an already recognized one, it must satisfy several criteria: First, it needs to be within a given range of the estimated position. Second, it needs to be the best possible position out of all recognized positions for the old one. And third, it must not be closer to an estimated position of another finger.

Calibration Issues

To use all the explained methods it is necessary to calibrate the system in a way that it can handle different settings and conditions. In our system there are two calibration procedures, one regarding the cameras themselves and one regarding the association between raw wall points and screen coordinates.

The camera calibration is necessary to allow different lighting conditions. In this procedure the important part is to adjust the threshold values of the binarization filter. With this filter, the system is able to separate background and fingers in a captured image during runtime.

The system needs to map raw wall points (given in real-world coordinates) to screen coordinates (given in pixels) by a transformation matrix. As our system has multiple displays embedded into the wall it needs to have a transformation matrix for every single display. Multiplying one of these matrices with a point v given in real-world coordinates results in a screen position v'. As the equation of $v' = A \cdot v$ must be true for every point on the wall we can use three arbitrary, linearly independent points to compute the matrix.

When the user has started the calibration mode, the system initially displays four points per display that the user needs to touch sequentially starting with the top left calibration point. After gathering a predefined number of intersections, the system will compute an average point to reduce errors during the detection process. The next step is to inform the user that s/he needs to touch the next calibration point. This is done by moving a circle from the previous to the next point. This procedure will continue until the real-world coordinates of all calibration points have been calculated.

Once we have the transformation matrix A, the system can start its calibration routine. In our case, the system takes four points including their known (x, y)-coordinates on a display. As mentioned before, the calibration only needs three points to work properly but we use the fourth point to minimize the errors of the detection during the calibration. Thus, the system computes four transformation matrices using paired dissimilar sets of three points. Finally it computes the mean value of each component of the matrix with the four temporary created matrices.

Fusion of multiple tracking data

An abstract input layer merges the gathered position data from the wall with the data received from the SmartBoard. Whenever a position has been detected by the SmartBoard, it will be added to a queue. The queue's length is two due to restrictions of SmartBoard as it can only detect two fingers simultaneously. If the queue contains an unprocessed finger event of a finger that currently has produced another event, the old event will be discarded. After the input layer has gathered the position data from both the wall and the SmartBoard, it will merge these different kinds of positions into a unified position representation. Later on, applications will not be aware whether a position has been tracked by the webcams or by the SmartBoard.

Before sending the information to listening applications, the input layer needs to match previously detected positions to newly detected ones. In the software architecture, each previously detected finger contains its current position, its movement vector, the amount of time it has been inactive, and two integers representing its current action ("*finger down*", "*finger move*" or "*finger up*") as well as its unique identification number. The reason for using a value of inactive time is that a finger might not be detected at all during one tracking cycle although it is still on the wall. This might happen if the finger moves too fast or if it is occluded by another finger in certain situations. After a specific amount time the finger has been inactive, it will be deleted from the array of active fingers and thus not be sent to listening applications.

Once all fingers have been updated, the input layer is ready to send the unified position data to all applications that have been registered for receiving input from the wall. As the input layer receives tracking data from both tracking devices in high frequency, it produces a steady stream of position data. Connected applications can use this stream analogical to the operating system's mouse input events.

6 EXAMPLE APPLICATIONS

We have implemented two example applications that make use of all input and output capabilities of our system. These applications use the steady input stream from the input layer as if they were system mouse events.

A Wall-sized drawing application

Our initial test application enables users to draw on the wall using their fingers. The three screens form one single display, but are tracked by different input systems. They represent the focus area of the application, since users draw on them. The steerable projector provides a tool palette upon request, which is displayed in the context area on the wall outside of the drawing area. This saves space otherwise wasted by placing tool bars in the drawing area. As mentioned before, the wall's size allows multiple users in front of it drawing simultaneously on the displays. Thus, the tool palette provides every user with their own choice of tools (line, curve, rectangle and eraser) independently. Users can "call" their palette by tapping on the wall outside of the displays. The palette will then be displayed at the detected position showing the currently selected tool for this user (associated via the nearest drawing position). Now the user can select another tool by simply touching it in the palette (see Figure 5).

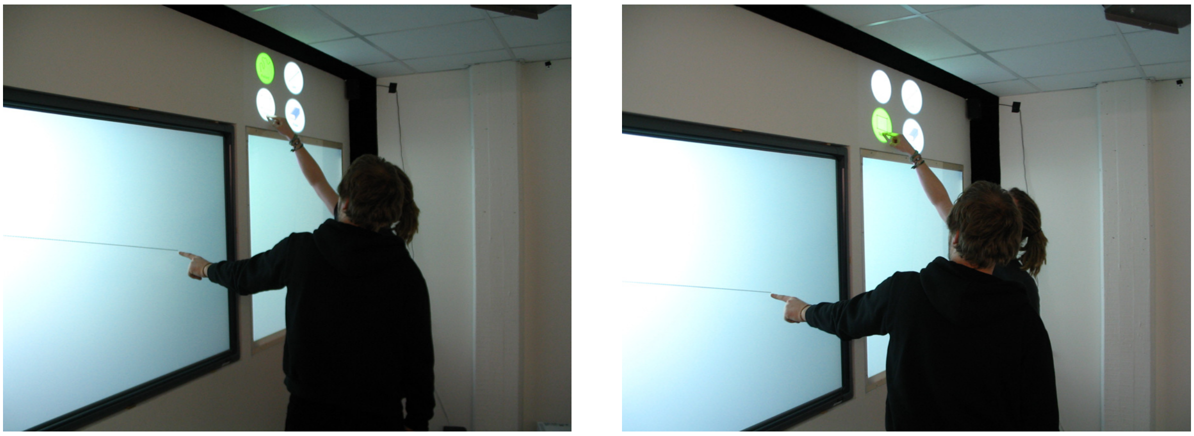

Figure 5: Left shows the displayed tool palette before selecting a new tool. Right shows it after the selection. Meanwhile, another user draws a line across the display wall.

BrainStorm

Our second and more complex demo is a brainstorming application for a team consisting of up to 4 users. The process of brainstorming is divided into different phases. In the first phase, all users comfortably sit around an interactive table (see Figure 6) and create notes of ideas by scribbling sketches or keywords on the desk's surface, which then turn into virtual post-it notes. Users interact on their side of the desk and write notes oriented correctly for them. When the notes are created, they appear simultaneously on the wall's focus display, where they are all oriented upright. Their relative positions on this display still match their positions on the table, so users can easily find their notes again on the wall display through a direct spatial mapping.

Figure 6: Overview of our room with the wall-sized display in the background, the steerable projector on the ceiling, and an interactive table in the foreground.

Users can now stand up and move notes around the entire wall, group them, form clusters by drawing a line around a set of related notes, and create connections among clusters and between clusters and notes by drawing a connecting line (see Figure 7). Notes or clusters, which are decided to be of secondary importance in this phase, can be moved to the context area of the display, i.e. the two side displays. Notes or clusters which are discarded altogether in the second brainstorming phase, can be deleted from the wall by dragging them over the border of a back-projection display, i.e., out of the working area.

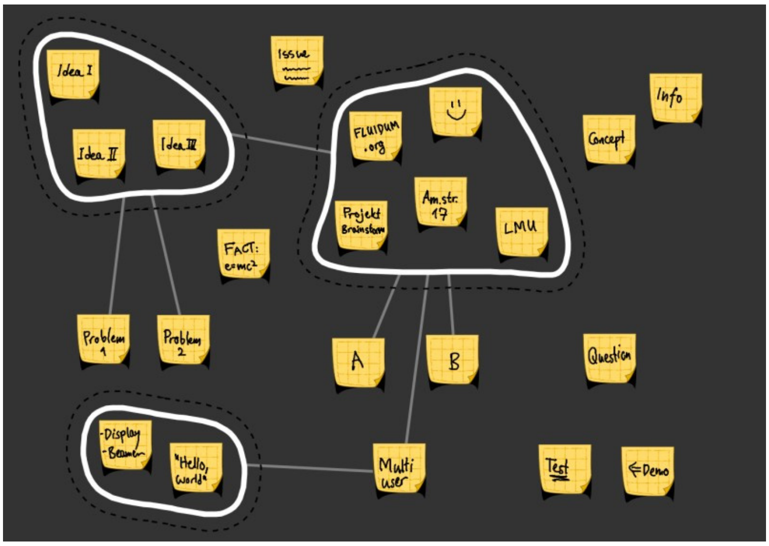

Figure 7: Screen shot of the focus display containing virtual post-it notes, which are partially organized.

In this setup, the inner focus display represents a group focus. It is still large enough to accommodate multiple user interaction, but only on this middle display, fine-grained manipulation takes place. The two adjacent context displays serve as a visual clipboard, where clusters can be stored when they are currently not being dealt with. This frees space in the focus area, but maintains visibility of these clusters, so that the whole collection of notes and clusters can still be overlooked when users take a step back. This second structuring phase eventually results in a mind map of related and grouped concepts and visually represents the result of the brainstorming.

7 CONCLUSIONS AND FUTURE WORK

We have presented a wall-sized focus plus context display using different tracking technologies and have shown two example applications of it. The display implements the focus plus context concept in two directions. In the output direction, it contains the back projection displays, which provide a simultaneous resolution of about 0.8 pixel per millimeter, and the rest of the wall which can display content using the steerable projector at a resolution of about 0.3 pixel per millimeter, but most importantly, only in one limited area of about 60x45cm at a time, i.e. time-multiplexed. In the input direction, the wall is split differently: the middle one of the back projection displays has pixel-exact low latency finger input, which is provided by the commercial SmartBoard hardware. The rest of the wall, including the adjacent back-projection displays and the wall above and below, has less fine-grained input at a resolution of about 10-20 mm, depending on the exact location and a tracking rate of 15Hz independent on the number of fingers. This divides the focus plus context display into three primal areas as shown in Figure 8.

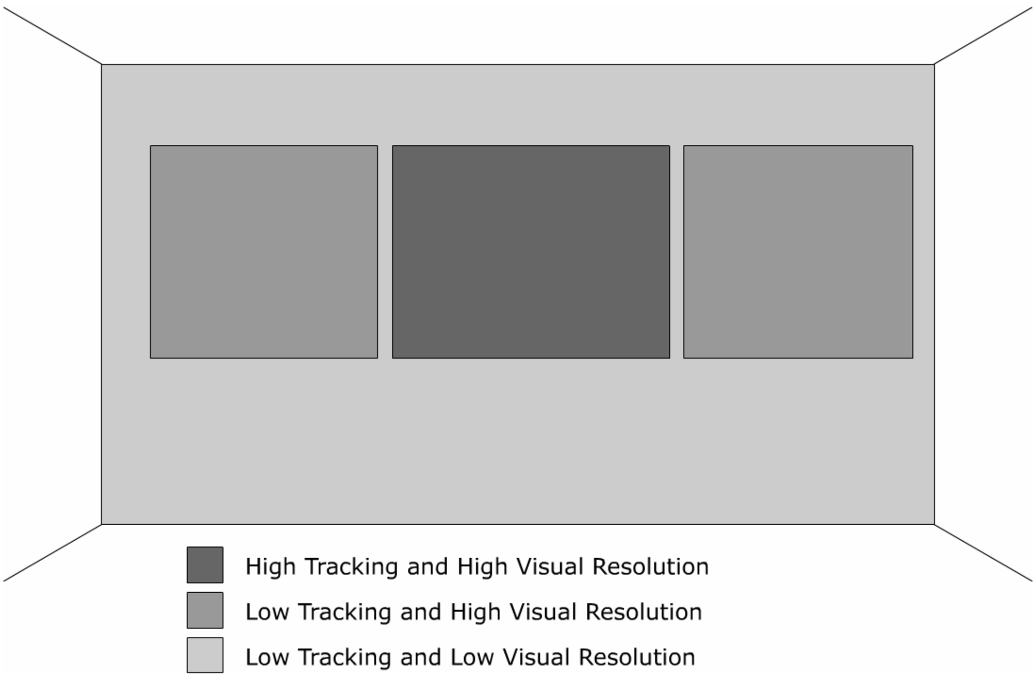

Figure 8: This shows our multi-level focus plus context display in both the input and output direction which enables a more fine-grained functional distinction between these areas.

Our demo applications describe a close co-located collaboration between several users on a single large display. Users share a common area of attention and interact simultaneously in it. Particularly in this situation, the distinction between different levels of focus makes sense because of the wide viewing angle we have to cover to see the full display.

Our current implementation still suffers from a slight jitter due to timing problems with the four USB cameras. When a finger is moving and the camera images are not taken at the exact same time, the computed positions are slightly wrong. We hope to increase tracking accuracy and reduce jitter in the outer tracking areas by applying temporal filtering to the signals.

ACKNOWLEDGMENTS

This work was funded by “Deutsche Forschungsgemeinschaft” (DFG). We thank Lucia Terrenghi for designing as well as David Kim for implementing substantial parts of the second demo application.

REFERENCES

1. Ballagas, R., Rohs, M., Sheridan, J. and Borchers, J., Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays. In CHI '05 extended abstracts, ACM Press, 2005, 1200-1203
2. Baudisch, P., DeCarlo, D., Duchowski, A., and Geisler, B. Focusing on the Essential: Considering Attention in Display Design. In Communications of the ACM 46(3), March 2003, 60-66
3. Baudisch, P., Good, N., Stewart, P., Focus plus Context Screens: Combining Display Technology with Visualization techniques, in Proceedings of UIST 2001, 31-40
4. Bezerianos, A., Balakrishnan, R. (2005). The Vacuum: Facilitating the Manipulation of Distant Objects. In Proceedings of CHI 2005, April 2-7, 2005, Portland, Oregon, USA
5. Cao, X., Balakrishnan, R., VisionWand: Interaction Techniques for Large Displays using a Passive Wand Tracked in 3D. ACM CHI Letters, 5(2).
6. Collomb, M., Hascoet, M., Baudisch, P., and Lee, B. Improving drag-and-drop on wall-size displays. In Proceedings of GI 2005, May 2005, Victoria, BC, 25-32.
7. Dietz, P., Leigh, D., DiamondTouch: A Multi-User Touch Technology, In Proceedings of UIST 2001, November 2001, Orlando, FL, USA, 219-226
8. Geißler, J., Shuffle, throw or take it! Working Efficiently with an Interactive Wall. In Proceedings of CHI 1998, ACM Press p.18-21
9. Han, J., Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection, In Proceedings of UIST 2005, 2005, Seattle, USA, ACM, 105-108
10. Igarashi , T., Freeform User Interfaces for Graphical Computing, In Proceedings of 3rd International Symposium on Smart Graphics, Springer LNCS 2733, 39-48
11. Johanson, B., Fox, A., Winograd, T., The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, In IEEE Pervasive Computing Magazine 1, 2002, 71-78
12. Küpper, A., Location-based Services: Fundamentals and Operation, John Wiley & Sons Ltd., Weinheim, Germany, 2002
13. Logitech Europe S.A., Website, http://www.logitech.com, 2006
14. Matsushita, N., Rekimoto, J., HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall, In Proceedings of UIST 1997, 1997, Banff, Canada, ACM, 209-210

15. Mynatt, E. D., Igarashi, T., Edwards, K., LaMarca, A., Flatland: New Dimensions in Office Whiteboards, In Proceedings of ACM CHI 1999, ACM Press, 346-353.

16. Pinhanez, C., Using a Steerable Projector and a Camera to Transform Surfaces into Interactive Displays, In Proceedings of CHI 2001, Seattle, WA, USA, March 2001, ACM, 369-370

17. Prante, T., Streitz, N. A., Tandler, P., Roomware: Computers Disappear and Interaction Evolves, In IEEE Computer, 2004, 47-54

18. Publictec Presentation Systems and Event Services, Website, http://www.publi.de, 2006

19. Rekimoto, J., Saitoh, M., Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments, In Proceedings of CHI 1999, 1999, Pittsburgh, USA, ACM, 378-385

20. Rensink, R.A., Internal vs. External Information in Visual Perception. In *Proceedings of 2nd International Symposium on Smart Graphics*, 2002.

21. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., Tan, D., Large Display User Experience. In IEEE Computer Graphics & Application, Special Issue on Large Displays, July/August 2005, 44-51

22. SMART Technologies Inc., SMART Technologies Inc., industry leader in interactive whiteboard technology, the SMART Board, http://www.smarttech.com/