Diploma Thesis - Final Report:
“A Wall-sized Focus and Context Display”

Sebastian Boring
Ludwig-Maximilians-Universität München
Agenda

- Introduction
- Problem Statement
- Related Work
- Design Decisions
- Finger Recognition
- System Setup
- Implementation
- Conclusions & Future Work
Introduction

» Instrumented Environments / Ubiquitous Computing

- Used to simulate ubiquitous computing in specific settings
- Equipped with sensors and other tracking devices

Questions:
- How can information be displayed in such environments?
- How does interaction need to be implemented to support users?
Problem Statement

- Large display wall (three back-projected displays + steerable projector)
- Interaction without augmentation of users
- Requirements:
  - Tracking of multiple fingers
  - Low-cost system (Webcams)
  - Complete wall as one display
Related Work

» Instrumented Environments (I)

- Interactive Workspaces Project (*iWork*) [7]:
  - Meeting and office room
  - *iROS* for communication
  - No interactivity on the whole wall

- *Roomware* [12]:
  - *DynaWall®, InteracTable®, ConnecTable® and CommChair®*
  - Limited to pen-based gestures
Related Work

*Instrumented Environments (II)*

- **Augmented Surfaces** [15]:
  - Computer augmented environment
  - Virtual extension of limited screen size (wall, tables, mobile displays)
  - Indirect interaction

- **Interaction Techniques:**
  - *Pick’n’Drop* [14] → Pen-based
Related Work

» Focus plus Context Displays

- Region of special interest (*focus*)
- Global view preserved (*context*)

**Focus plus Context Screens** [1][2]:

- Combination of display and visualization techniques
- Fixed focus region
- Interaction only possible in the focus region → context provides overview
Related Work

» Touch Sensitive Displays (I)

- **HoloWall [10]:**
  - Rear-projected glass wall
  - Tracking: IR-Camera

- **FTIR [5]:**
  - Basic idea of fiber optics
  - Acrylic surface with IR-LED arrays attached to it
  - Tracking done by rear-mounted IR-Camera
Related Work

» Touch Sensitive Displays (II)

- SMART Technologies [18]:
  - IR-LEDs in the frame surrounding the screen
  - IR-Cameras mounted in the corners
  - Angulation as tracking method

- DiamondTouch [4]:
  - Capacitive sensing
  - Users are actively “connected” to the computer
Design Decisions

» Hardware Decisions

- The Display Wall:
  - Three displays (SmartBoard in the center, projection foil on the side)
  - Steerable projector (beamMover 40 by publitec [13])

- Tracking System:
  - SmartBoard for high resolution
  - Webcams (Logitech QuickCam Fusion [9]) for low resolution
Design Decisions

Software Decisions

- C# vs. Java:
  - C# supports multiple cameras with DirectShow (Java/JMF only supports one camera at a time)
  - Java for communication (Event Heap [6]) and visualization (Swing)
- SmartBoard SDK for input events
- C/C++ DLL (with JNI) for the steerable projector
Finger Recognition

- The Tracking Method (I)

- Lateration:
  - **Range (Difference):** Traveling time (difference) of emitted signal

- Angulation:
  - **Antenna arrays (sectorized)**
  - **Each pixel is an antenna**

Circular lateration (2D) [8]

Hyperbolic lateration (2D) [8]

The basic principle of angulation (2D) with two receivers. The positions (X, Y) are known before the measurement of the angles [8]
Finger Recognition

» The Tracking Method (II)

Hyperbolic lateration (2D) [8]

Circular lateration (2D) [8]
Finger Recognition
» Calculating Angles from Live-Captured Images (I)

- **Frame Differencing:**
  - Does not work if screens are turned on → Fingers will have the same color as the background

- **HSL Filtering:**
  - Does not work if screens are turned on → Fingers will have arbitrary colors

- **Non-reflection surface needed**
  → black velvet
Finger Recognition

» Calculating Angles from Live-Captured Images (II)

Different filter techniques for object (e.g. finger) recognition. Top left shows differencing, top right shows HSL filtering. Bottom illustrates differencing with black velvet mounted onto the wall.
Finger Recognition

» Calculating Angles from Live-Captured Images (III)

Geometrical illustration of the calculation of images captured by a camera.
Finger Recognition
» Process of Triangulation (I)

The positioning using angulation with four receivers. The magnification illustrates the error potential for a single position. Small circles indicate intersections of a pair of lines from two receivers.
Finger Recognition
» Process of Triangulation (II)

- Multiple fingers require more than two receivers
- Calculate the line between each detected finger and every camera
- Take two cameras and intersect all detected lines
- Intersection is valid if it is close enough to the lines of the other two cameras
Finger Recognition

» Continuity of Finger Movement

- Recognized fingers need to be matched to previously detected fingers

- Dead Reckoning as simple method:
  - Store the orientation vector of the last movement
  - Make the assumption that the finger moves in the same direction
  - Old position + orientation vector = estimated position
Finger Recognition

» Calibration Issues

- Camera Calibration:
  - Individual settings for each camera
  - Calibration image, camera’s position and filter threshold as parameters

- System calibration:
  - Association between raw position data and screen coordinates
  - Interactive calibration procedure
System Setup

The Display Wall

Wall setup and display arrangement. Unit of measurement is millimeter.
System Setup

» Camera-based Tracking System

Different views of the fixation for one camera.

Design drawing of the fixation for one camera. Unit of measurement is millimeter.
System Setup

» Complete System Overview

Basic setting of the system in the instrumented room. [3][9][16][17]
Implementation

» The Tracking Engine

- Implemented in C# + DirectShow
- Calculates raw positions and sends it to listening applications through the Event Heap

Screenshot of the main application window and the detection window. In the detection window, the user is able to see currently detected objects which will be drawn as lines in the main application window.
Implementation

» Camera Calibration

- Implemented in C# + DirectShow
- Stored calibration data in XML

Screenshot of the device configuration window and the filter configuration window. Both give the user all necessary controls to calibrate and adjust device related data.
Implementation

Interactive System Calibration

- Runs on two different machines:
  - Tracking engine (C#)
  - Calibration display (Java)

The calibration screen projected onto the displays and bottom right demonstrates a user calibrating the system.

Sketch of the calibration screen (projected onto all three displays) with the first point as active point.
Implementation
» Connecting Fingers and Positions

- **Input Layer:**
  - Merges input from the wall and the SmartBoard
  - Associates recognized fingers with previously detected ones
  - Sends unified finger positions to listening applications
Implementation

» Steerable Projector

- Runs on a third computer
- Provides an interface to the USB2DMX interface to control the projector
- Available DMX [19] values:
  - Pan/Tilt
  - Focus/Zoom
  - Trapeze adjustment (H/V)
  - Control channel
Conclusion & Future Work

» Possible Improvements of the System (I)

- Jitter as a major problem

Example of a detection process with the currently used method. The numbers represent the time stamps of each line.
Conclusion & Future Work

» Possible Improvements of the System (II)

Possible solution to avoid jitter:

Example of a detection process with the interpolated method. The numbers represent the time stamps of each line.
Conclusion & Future Work

» Possible Improvements of the System (III)

- Low resolution is not able to detect closely neighbored fingers:
  - Fingers might be undetected
  - Two positions might be merged into one (inaccurate)

- Traces of fingers can be incorrect with the current method of dead reckoning in special cases
This shows two positions on the wall where only one will be detected. The upper magnification shows the detection error due to a too large distance of the calculated position from other lines. The lower one shows the correct detection of the second point on the wall.
Conclusion & Future Work

- Possible Improvements of the System (V)

This shows the inaccuracy as two intersections existing on the wall are merged into one detected intersection, causing jitter over a given time span.
Conclusion & Future Work

» Overall Impression

- Inexpensive wall-sized focus plus context display
- Execution speed $\leq 100$ ms for each detection cycle
- Several input- and output technologies combined that form a large interactive display
- Not restricted to a single user
  $\rightarrow$ simultaneous detection of up to four fingers possible
Conclusion & Future Work

Future Work

- Improvement of the tracking system in the near future
- Image equalization for the steerable projector to display rectangular images
- Implementation of a large demo to demonstrate the instrumented room
Thank you!

Sebastian Boring (sebastian.boring@ifi.lmu.de)
Ludwig-Maximilians-Universität München
Bibliography (I)


Bibliography (II)


Bibliography (III)


17. Shopping.com, *Siemens Scenic X102 (FAPSCED10202GB) PC Desktop*, http://img.shopping.com/cctool/PrdImg/images/pr/177X150/00/01/60/8c/79/23104633.JPG

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

UML-Diagrams

» Device System
UML-Diagrams

» Event Heap Communication

Event Type: DisplayWallPositionSet
Fields:
- WallPositionX1: String
- WallPositionY1: String
- PositionX1: String
- PositionY1: String
- Inaccuracy1: String
- RegionID1: String
- WallPositionX2: String
- RegionID2: String
- WallPositionX3: String
- RegionID3: String
- WallPositionX4: String
- RegionID4: String

Time to live [ms]: 100

Event Type: NewPositions
Fields:
- FingerID: String
- Action: String
- PositionX: String
- PositionY: String
- WallPositionX: String
- WallPositionY: String

Time to live [ms]: 100

Event Type: CalibrationEventSet
Fields:
- Command: String

Time to live [ms]: 100

Input Layer

Tracking System

Event Heap

DisplayWall Calibration

Application

München, 26.04.2006
Sebastian.Boring@ifi.lmu.de
UML-Diagrams

» Event Heap Connection
UML-Diagrams
» Graphical User Interface
UML-Diagrams

» Mathematical Implementation
UML-Diagrams

» Input Layer
UML-Diagrams
» Internal System Events (C#)
Flow Charts
» Position Calculation

Start

Get Detected Lines

Detected Lines

One Line per device available?

Intersect Lines

Intersection

Intersection valid?

Add to List

More lines available?

Send Intersections

München, 26.04.2006

Sebastian.Boring@ifi.lmu.de
Flow Charts
» Transformation Matrices

Start

Transform position with center matrix

Is in left display?

yes → Transform position with left matrix

no → Is in right display?

yes → Transform position with right matrix

no → Transformed position

End