The mother of all disco floors

T. Delbrück[1], A.M. Whatley, R. Douglas, K. Hepp, P.F.M.J. Verschure

Institute of Neuroinformatics[2], ETH/Univ. Zürich, Switzerland

Abstract: We describe the floor used as the skin of the playful interactive space *Ada*, which was part of the Swiss National Exhibition Expo.02. The floor consisted of 360 hexagonal tiles, each of which comprised an aluminum frame, a glass top, a microcontroller, analog load sensors, and dimmable red, green, and blue neon lamps. A factory automation bus sensed and controlled the tiles. We developed software for signal processing of the load information, for generating visual effects on the floor, for tracking visitor paths, and for a variety of games and interactions.

Keywords: interactive space, Ada, neon, load sensor, force sensitive resistor, FSR, people tracking, dance, disco, factory automation, Interbus, game

[1] tobi@ini.phys.ethz.ch
[2] www.ini.unizh.ch

1 Introduction

'Ada' was an intelligent space that interacted with its visitors using touch, audition, sound, and vision. You can think of Ada a bit like a playful robotic disco; its world was its visitors. The floor was the principal medium for interaction. Here we will start by briefly indicating how Ada used the floor to interact with the visitors, and then discuss the design of the physical floor and its software[3].

Ada used the its floor to track visitors and to interact with them. In the simplest interaction, Ada reacted to being stepped on by generating a transient visual effect like a surrounding ring of tiles that would light up and then slowly fade away. In a more complex interaction, individual visitors were tagged with a certain tile color that they carried with them as they were tracked over the floor. In Figure 1, a man has captured Ada's attention and is surrounded by a ring of lit tiles that follow along as he moves around.

Ada also used its floor to actively measure a visitor's willingness to interact. Tracked visitors were presented with a flashing tile just next to them. If they stepped onto the flashing tile, the flashing effect moved to a neighboring tile. If the visitor followed the flashing tile for a few steps, the space considered that person compliant[4]. Compliant visitors received extra attention; they saw a pulsating ring of tiles around them, *light fingers*[5] pointing at them, and *gazers*[6] looking at them: in short, they were made to feel special.

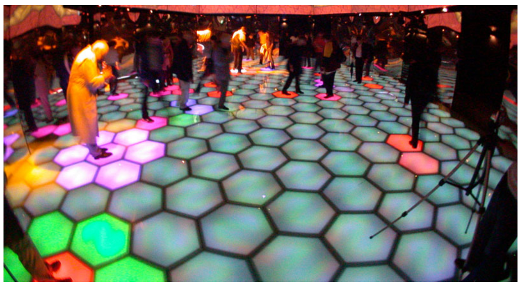
Figure 1 Ada in operation. The man at the left has captured Ada's attention.

Ada also used the floor to play games with the visitors. In the most commonly used game, visitors chased and tried to stomp on a virtual ball (a brightly lit white tile). This virtual football skittered about, bouncing off the walls and visitors, while doing its best to evade visitors. A successful stomping resulted in a victory reward for the winner; they were surrounded by a halo of light that grew and then faded away; they were highlighted by light fingers and targeted by gazers; their image was placed on the *big screen*[7] for all to admire.

Many luminous floors have been constructed by the entertainment industry, for use in discotheques, TV studios, and stage shows. These floors enable remote control of the lamps, but they do not sense and react to people. What about tactile floors? (Morishita, Fukui, & Sato 2002;Orr & Abowd 2000;Taketoshi & Tomomasa 1998), and

[3] For general information about Ada see the web page www.ada-exhibition.ch, or (Eng et al. 2003). To experience Ada's behavior see the companion video paper (Delbrück et al. 2003).
[4] Think of a dog holding a stick in its mouth and wagging its tail at you, testing if you want to play.
[5] Ada's fingers: steerable theater lights.
[6] Ada's eyes: pan tilt cameras.
[7] An enclosing ring of video projection screens.

Selker's group at MIT Media Labs have built tactile floors that can determine the locations of people or their feet, interaction, however, is left to other components of the installation.

2 Tile design

Ada required a floor that was both tactile and luminous, so that Ada could reliably track its visitors and interact with them. The scale of the project required networking instead of dedicated cables to each sensor or tile. We needed an industrial-strength floor that could stand up to thousands of people per day for many months of operation; we were bound contractually to keep downtime to below 3%.

Figure 2 shows the inside of one of the tiles.

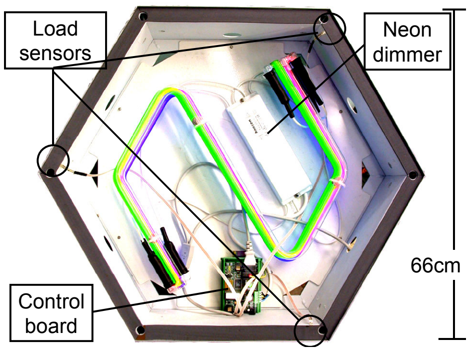

Figure 2: The inside of the tile.

2.1 Networked control

After a long struggle with our own notions of how the tiles could be networked using cellular automata, we settled on a much simpler daisy-chained factory automation network called Interbus[8].

Interbus has been widely used since the mid-1980's[9]. It has several features that make it suitable for use in our floor: It is a master/slave bus; a single PC with several Interbus master boards can control the entire floor. It is daisy chained, which greatly simplifies floor installation. Chips implementing the protocol are available and documented. Each node is sensed and controlled at predictable intervals. It is good for automation of devices with a small amount of sensor and control data. It is self-configuring, and has robust error checking and diagnostic capabilities.

[8] www.interbusclub.com

[9] murray.newcastle.edu.au/users/students/1999/c9518176/interbus.html is a readable history of these fieldbuses.

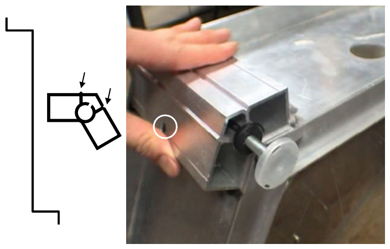

Figure 3: Tile frame profile shapes (not to scale) and outside corner of frame. Arrows show the mating alignment extrusions. The circle shows one of the holes for bolting the tiles to each other.

2.2 Interbus master interface

A daemon-like interbus process uses Linux drivers for the Hilscher[10] Interbus master controllers to communicate between the Interbus master controller boards and the floor-controlling software. A shared-memory interface allowed development of the higher-level functionality to be decoupled from the hardware. To higher-level processes, the floor tiles appear as a set of shared memory segments, one representing the load values of the tiles, another the colors to be displayed on the tiles. An arbitrary number of processes on the controller PC can read from the tile load and temperature[11] segments. A single process can write to the colors segment. Ada's floor ran at an update rate of about 15Hz.

2.3 Neon lighting

Neon lighting is a mature technology. It is reliable and very power-efficient. It is not as trendy as LED illumination, but the technologies for production and assembly are known to many lighting firms like Westiform[12], the firm we contracted to build the production version of the floor.

Three neon lamps—red, green, and blue—illuminate the tiles. Westiform shaped the neon tubes in the sigmoid form shown in Figure 2. The lamp brightnesses were controlled by commercial 3-channel 80mA, 990V neon dimmers[13] that were customized to accept control over a RS232 link. Each dimmer cost about $200.

2.4 Physical construction

The design of the tile frames was important because the floor had to withstand heavy traffic over months of operation. We also

[10] www.hilscher.com

[11] Tile temperature was monitored for safety; it never rose above 45°C.

[12] www.westiform.com/homepage

[13] www.toni-maroni.de

wanted to be able to reinstall the floor at another location relatively easily.

Figure 3 shows how Westiform built the tile frames from extruded aluminum. They used two extruded aluminum shapes. One piece forms the walls of the frame and the other forms the legs. Adjustable feet are press-fitted into the legs. The tile frames are bolted to each other. A ridge and indentation on the leg pieces register the tiles to each other resulting in a very stable structure.

The tile tops are made from tempered and laminated safety glass. Each of the two plates is 8mm thick. Translucent PVB is used to bind the plates, and an additional 3mm translucent polycarbonate beneath the glass top further diffuses the light to better mix the colors. Each top costs about $100.

2.5 Tile controller

A local slave controller in each tile (Figure 4) reads the sensors, controls the neon dimmer to set the lamp RGB brightness, and communicates with the Interbus. It also enables self-test and automatic sleep mode.

The tile controller uses a Mazet IB8052[14] microcontroller with embedded Interbus data-link hardware. The firmware is about 800 lines of C code; it occupies 11kB of EPROM space.

2.6 Load Sensors

We used FSRs (Force Sensitive Resistors) as our load sensors after considering and rejecting several alternative technologies. FSRs are flat devices. We mounted 3 of these equally spaced at the corners of the hexagonal tile frame between the frame and the glass tile top, under a 3mm layer of EPDM rubber. Their conductance increases monotonically with the applied load, approximately as a square-root relationship.

The tile measures the load on its glass top by forming a voltage divider with each of the three FSRs (Figure 4). The master controller combines the three FSR values arithmetically to form the tile load value. This value is monotonically related to the load applied to the tile.

The tile-to-tile variation in DC load output is nearly half of the full-scale value. Manufacturing differences in the tile frames cause most of this huge mismatch. The excitation of the neon tubes by the 1kV, 20kHz voltage pulse causes a huge impulsive noise spike if a neon excitation pulse occurs while a nearby load sensor is being read.

[14] New designs can use any microcontroller that can communicate with an Interbus SUPI3 interface chip, sold by Phoenix Contact (www.phoenixcontact.com) for about $5.

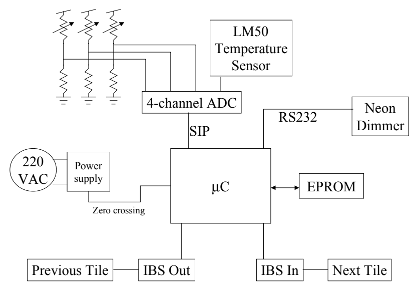

Figure 4: Functional components of floor tile slave controller board.

The tile load sensor mismatch and impulsive noise required that we filter the raw load signals before we determining whether tiles were loaded. These filtering operations were done on the master controller. We used a median filter to remove the impulsive noise. We used a nonlinear high pass filter to remove the DC component of the tile load. The floor controller learned an optimum DC value for each tile that is subtracted from the raw load to produce the filtered load; a fixed threshold then determined whether the tile was loaded. The learned DC load (which estimates the unloaded raw load of the tile) adapted slowly upwards while the tile was loaded, and rapidly recovered when the tile was unloaded. The reason for the different adaptation time scales for loaded and unloaded states was to slowly forget loaded states and rapidly recover from unloaded states.

3 Floor control software

Ada's software is a heterogeneous mixture of procedural code and neural networks. Most of the procedural code, including nearly all the floor code, was written in Java. We chose Java because of its high productivity, debugging tools, and remote procedure support. Hard real time performance was not critical for this exhibit, but we were pleased by Java's ability to deal with a huge, soft real-time environment.

A single PC handles the entire floor. It runs the C++ interbus process and a Java floor server process comprising about 80 Java classes with about 20,000 lines of code.

3.1 Rendering

The usual 2-dimensional image rendering surface is not appropriate for managing the colors of Ada's tiles. Nearby tiles dominate the visitor's view. We developed methods for rendering fluid patterns of activity that could be locked to a tile or to a tracked visitor. We also developed a general set of dynamic patterns that could be displayed on floor regions. These local and global effects were the only outputs onto the floor.

Created effects are placed in a list and are updated every floor cycle until they expire or are removed. Each effect knows how to update itself, how to compute the tiles that should be affected, how to set their brightness, and so forth. A large number of parameters control the dynamics of the effects.

Reactive effects that were automatically created on freshly loaded tiles constituted most of the behavior of Ada during its simpler behavioral modes—like sleeping or encouraging visitors to leave.

3.2 Tracking

The primary objective of Ada was to identify individuals and interact with them. To enable these interactions, Ada had to track individuals over time, so that a label assigned to them—a special color or pattern—could travel along with them, or so that a gazer or light finger could follow them.

Our tracking algorithm is relatively simple; our development timeline did not allow for great sophistication. Nonetheless, it does a credible job in tracking visitors. It easily runs in real time. It does particularly well when the space is not crowded (<10% loaded tiles) and when the visitors are cooperative, in the sense that they want to be tracked. Such visitors usually step on single tiles and keep a polite tile distance from each other. By contrast, children run wildly through the space trying to step on as many other people's tiles as possible. When the space is relatively uncrowded, naive visitors are tracked fairly reliably.

The tracking algorithm is applied during each load sensor update cycle. As an initial step, the filtered weight sensor values (as described in 2.6) determine whether a tile is loaded or not. All tracking is based on these binary tile states.

When a tile assigned to a tracked person becomes *unloaded*, nearby loaded tiles that have not been loaded for too long are assigned as *possible* destinations of that person. A list is built of *all* possible destinations by *all* tracked people. This list is pruned by matching persons to destinations. As each match between target and destination is made, the corresponding objects are removed from further consideration.

The result of the tracking algorithm is a list of tracked objects maintained by the floor server process. These objects are proxies for person-objects that are maintained on a separate object server process that deals with all objects that Ada knows about. The floor server only transmits changes in the proxy list to the object server. The operation of the object server and the behaviors it generates are not discussed here.

4 Conclusions

To our knowledge this is the first floor that has been built that is both luminous and tactile, and that provides for individual tracking of multiple persons. The high cost ($800/tile) of the specially-produced present floor limits its applications, but Ada demonstrated that it could be used reliably for fairly sophisticated public interaction with >500,000 naive visitors over 5 months of continuous operation.

5 Acknowledgements

G. Ritler constructed the floor tile bases for the prototype floors. Westiform's expert design, management, and construction (E. Lüdi, M. Gloor, and H. Peyer) enabled the final form of the floor. O. Hasselbusch (OHTec) assisted in design of the production control board. We thank U. Rutishauser, K. Eng, M. Erzinger (ETH Zürich), and R. Derrer (Delux AG) for valuable assistance.

Ada was funded by the University and ETH Zürich, Manor AG, and the Velux and Gebert Rüf foundations. Additional sponsorship for Ada's floor came from Westiform AG and Toni Maroni GmbH.

References

Delbrück, T. et al. 2003, "Ada: a playful interactive space", Interact 2003.

Eng, K. et al. 2003, "Design for a brain revisited: The neuromorphic design and functionality of the interactive space Ada", *Reviews in the Neurosciences*.

Morishita, H., Fukui, R., & Sato, T., "High Resolution Pressure Sensor Distributed Floor for Future Human-Robot Symbiosis Environments", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland.

Orr, R. J. & Abowd, G. D. 2000, "The Smart Floor: A Mechanism for Natural User Identification and Tracking", in CHI '00 Extended Abstracts on Human factors in computer systems (CHI 2000)), The Hague, Netherlands, April 1-6, 2000, ACM Press, New York, pp. 275-276.

Taketoshi, M. & Tomomasa, S. 1998, "Robotic Room: Its concept and realization", *Robotics and Automation*, vol. 16, no. 5, pp. 705-717.