Available online at www.sciencedirect.com

ScienceDirect

Robotics and Autonomous Systems 55 (2007) 433–443

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

A tactile luminous floor for an interactive autonomous space

Tobi Delbrück*, Adrian M. Whatley, Rodney Douglas, Kynan Eng, Klaus Hepp, Paul F.M.J. Verschure

Institute of Neuroinformatics, Winterthurerstr. 190, University of Zürich and ETH Zürich, CH-8057 Zürich, Switzerland

Received 24 March 2006; received in revised form 17 January 2007; accepted 25 January 2007
Available online 22 February 2007

Abstract

This paper describes the interactive tactile luminous floor that was constructed and used as the skin of the playful interactive space Ada, which ran as a public exhibit for five months in 2002 and had over 550,000 visitors. Ada's floor was custom-built to provide a means for individual and collective user interaction. It consists of 360 hexagonal 66 cm tiles covering a total area of 136 m^2, each with analogue tactile load sensors based on force-sensitive resistors and dimmable neon red, green and blue (RGB) lamps. The tiles are constructed from extruded aluminum with glass tops. An Interbus factory automation bus senses and controls the tiles. Software is described for rendering fluid, dynamic visual effects on the floor, for signal processing of the load information, for real-time visitor tracking and for a variety of behavioural modes, games and interactions. Data from single tiles and from tracking are shown. This floor offers new modalities of human–computer interaction and human–robot interaction for autonomous robotic spaces.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Interactive robotic space; Tactile surface; Luminous floor; People tracking; Computer games

1. Introduction

Interactive floors that can both sense the presence of persons and provide output in the form of light have a wide range of potential application areas, including entertainment, surveillance, healthcare, building safety and pedestrian guidance systems. The key to their attractiveness for interactive spaces is the co-location of the sensors and effectors, providing an intuitive visual interface for non-expert users. Although this idea has probably been considered repeatedly over the last few decades, the complex task of constructing such a floor has meant that there are no instances of interactive floors that are presently in real-world, everyday use. The six key challenges in building a feasible interactive floor are:

1. providing good-quality person detection;
2. ensuring physical and electrical robustness as well as maintainability and user safety during continuous real-world operation;
3. robustly communicating bi-directionally with a large number of the floor tiles;
4. generating highly visible, aesthetically pleasing, controllable illumination;
5. developing a scalable and reusable control software infrastructure, and
6. developing captivating and intuitive visitor interactions.

To date, the efforts in this field perform well in one or two areas, but not in all of them. Luminous floors are commercially available from at least five suppliers for use in discotheques, television studios and stage shows; these floors enable remote control of the lamps but have no tactile capability. One company supplies a LED-based luminous tactile floor, but no specifications are publicly available and interactions appear to be limited to reactive effects (www.lightspacecorp.com).

A handful of prototype tactile floors without visible light output have also been constructed: References [1–7] report tactile floors that can determine the locations of people or their feet with relatively high temporal or spatial resolution – greater than 50 Hz and down to a few centimeters – which were developed for musical instrument input, automated dance choreography and health-monitoring activities. For the most

---

* Corresponding author. Tel.: +41 44 635 30 38; fax: +41 44 635 30 53.
E-mail address: tobi@ini.phys.ethz.ch (T. Delbrück).
URL: http://www.ini.uzh.ch/~tobi/ (T. Delbrück).

0921-8890/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2007.01.006

Table 1
Comparison of tactile floors

Tactile floor	Date	Total area (m^2)	# tiles	# sensors	Spatial resolution (cm)	Temporal bandwidth (Hz)	Load resolution	Technology	Scalable?
PodoBoard [1]	1991	?	?	?	2.5	>100 Hz	binary	Electrical contact	?
MIDI controller [2]	1995	6	64	64 (16 × 4)	15	?	?	FSR	?
Magic carpet [3,8]	1997	5.6	–	16 × 32 crossing wires	10	∼0.5	∼3 bits?	Piezoelectric cables	?
Litefoot [4]	1998	1.8	121	1 936 (44 × 44)	4	50	binary	Passive photodetectors	?
Robotic room [5,7]	2002	4	16	65k (16*4096)	0.78	?	binary	Electrical contact in tactile PCBs	?
Ada's floor	**2002**	**136**	**360**	**1080**	**66**	**15**	**Tile: ∼5 bits** **Floor: ∼3 bits**	**FSRs**	**Yes**

part [1–7] report on technology for tactile sensing rather than visitor interaction. The largest of these floors covers an area of a small bedroom.

Four of the prior tactile floors were designed for musical instrument or dance control and are reviewed by Griffith [4]. Johnstone's PodoBoard (1991) [1] uses electrical contact over a grid of conductors. Dancer's shoes complete electrical connections between toes and heels. Pinkstone's MIDI controller (1995) [2] uses a 16 × 4 array of approx. 0.1 m^2 force sensitive resistors bonded to heavy plastic with a covering of polyethylene foam. A microcontroller reads the sensors and interfaces to a MIDI control unit. Paradiso's Magic Carpet (1997) [3,8] uses piezo-resistive wires arranged in crossing grids to sense foot pressure. It is limited like normal touch screens in that it can only reliably sense a single connected area. Griffith's Litefoot (1998) [4] is based on photosensors at each location that can detect changes in light intensity caused by shoe soles covering the sensors. Orr's single tile (2000) [6] uses commercial load cells sampled at 500 Hz to record detailed footstep pressure profiles, with the aim of identifying individuals who walk across the tile. Morishita's Robotic Room floor (2002) [5,7] is a high resolution binary load sensing tactile floor based on tiles with crossing arrays of wires patterned on large printed-circuit boards. The construction of the two layers of PCBs and their spacers enables detection of electrical contact covering regions of a few cm^2. Table 1 compares metrics of prior tactile floors with those of the floor reported here.

Ada [9–14] is a playful autonomous robotic space intended to stimulate public debate on the future of brain-like machine intelligence. It engages with its visitors using touch, audition, sound, and vision. We think of Ada as a robot turned inside out, with its "world" being its visitors. Ada ran as a public exhibit in the summer of 2002 as part of the Expo.02. It operated 12 h a day for five months and had 550,000 visitors. Ada is presently maintained as a much smaller version in our laboratory.

In developing Ada, we faced the challenging problem of developing a playful space that could interact individually and collectively with many people who moved about freely. It soon became clear to us that floor-based interaction was the most promising means of achieving this. After building prototypes that tried unsuccessfully to use video tracking, we decided that a tactile luminous floor would offer a better foundation for achieving reliable and effective interaction. A tactile floor gives Ada a reliable and low-latency sense of the immediate locations and footstep actions of individual visitors, and a luminous floor provides a means of interacting with individuals and with groups. No such floor was available for purchase, nor could we find any reference to previous work along these lines, so we had to develop this technology from scratch. This development took about three years and involved hardware, manufacturing and software engineering, with the real-world constraint that the solution had to be robust enough to stand up to massive numbers of visitors and reliable enough to function with an uptime mandated in our contract with the exhibition organizers of over 98%.

Ada's floor is the principal medium for interaction between the space and its visitors. In this paper we discuss the important hardware and software design characteristics of Ada's floor and indicate how the floor is used in Ada for visitor interaction. Section 2 introduces Ada's floor-based interactions, Section 3 discusses the design of the physical tiles, and Section 4 describes the floor software. Section 5 describes the games we developed to run on the floor. Section 6 concludes with a discussion of the implications of this development.

2. Ada's floor-based interactions

Ada was intended to give people the impression they were dealing with an inquisitive creature with distinct behavioral states. Ada's overall infrastructure was quite complex to enable a large variety of audio-visual visitor interaction: The space consisted of an elongated octagon, surrounded by a wall of mirrors. Above the mirrors, a 360 degree wall of projection screens (the *big screen*) displayed dynamic visual feedback to the visitors, including an animation of the overall behavioural state. Hanging from a frame above the space additional components of Ada enabled further visitor interaction. These consisted of two sets of three microphones (used for sound localization and word recognition), eight pan-tilt cameras (*gazers*, used to extract visual information from the space), four wide angle ceiling cameras (used for monitoring the space) 36 speakers (used for localized sound effects), a PA system (for soundscape) and 24 steerable theatre lights (*light fingers*, used to highlight visitors). The space was controlled by a cluster of 36 PCs running Linux.

During the exhibit, tour guides would allow visitors in groups ranging in size up to 35 people into the space. During the next 6 min, the visitors would experience a range of behaviours

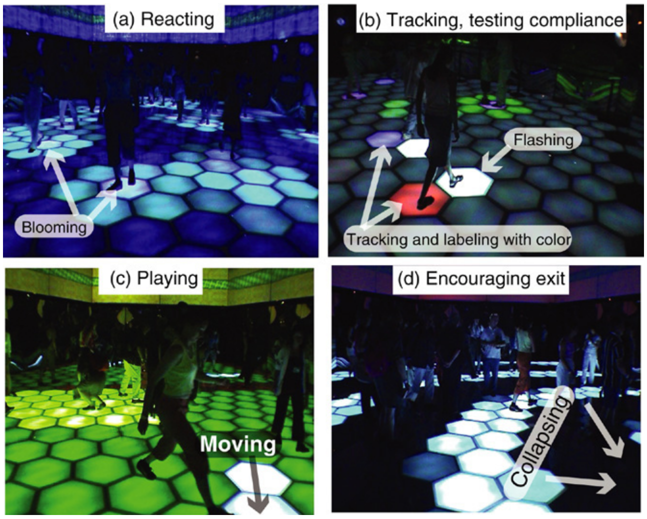


Fig. 1. Ada's floor in operation. (a) Ada sleeps and indicates it is tickled by flowers which bloom around newly loaded tiles; (b) tracked people are labelled with individual colours and are tested for their compliance; (c) a person chases a virtual ball, trying to jump onto it; (d) people are encouraged to leave.

analogous to those of a creature that is sleeping, is woken up, interacts by trying to engage visitors, plays a game with them and then becomes tired and wants to be alone again (Fig. 1). Visitors learn that they can tickle a sleeping Ada in a simple reaction when newly loaded tiles generate a transient visual effect, like a surrounding ring of tiles that slowly lights up and then fades away (Fig. 1(a)). Ada uses the rate and quantity of "tickle" reactions produced by visitors to judge when to "wake up" by turning the floor a bright yellow. Next, visitors learn that Ada knows about them as distinct entities in a more complex interaction, when they are tagged with an individual tile colour that they carry with them as they are tracked (Fig. 1(b)). Not all visitors pay attention to Ada. To find out on whom to spend extra resources, Ada uses its floor to actively probe a visitor's willingness to interact; the metaphor being a dog that holds a stick in its mouth and looks at you while wagging its tail. Tracked visitors see a flashing tile next to them which is first presented in the direction the visitor has been moving so that it is more likely to be noticed (also Fig. 1(b)). If the flashing tile is stepped on, it moves to a neighbouring tile. If the visitor follows the tile for a few steps, the space considers that person *compliant*. Compliant visitors are rewarded for their attention: they see a pulsating ring of tiles around them, light fingers pointing at them, and gazers looking at them. The gazers and light fingers are coordinated with floor tracking information to present a live video view of the highlighted visitor that dynamically moves along the surrounding big screen to appear in the direction of visitor motion, where they are most likely to notice it.

Ada also uses the floor to play games with visitors. In the most commonly used game, "Football", visitors chase and try to jump on a virtual ball that is indicated by a brightly lit white tile (Fig. 1(c)). The virtual ball skitters about, bouncing off the walls and the visitors and producing appropriate "bing" and "bong" sound effects. Visitor collisions increase the speed of the ball. Successfully jumping onto the ball results in a victory reward; winners are surrounded by a halo of light that grows and fades away.

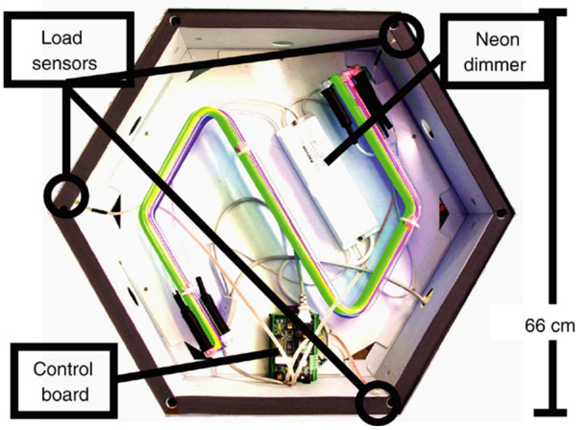


Fig. 2. Inside a single floor tile.

After playing this game with Ada for about a minute, the group is led to believe that Ada has tired and wants to be alone again: waves of light moving towards the exit direct visitors out of the space, and the cycle repeats again with the next group of visitors (Fig. 1(d)).

3. Tile design

To enable these interactions, some of which require tracking the visitors, Ada required a floor that was both tactile and luminous. The scale of the project required networking rather than dedicated cables to each sensor or tile. Ada also required an industrial-strength floor that could stand up to thousands of people per day for many months of operation.

The final form of Ada's floor tiles was the result of about three years of development of three major prototypes. These were constructed first from wood and Plexiglas using binary 12 V halogen lamps with dedicated cables, then from wood and Plexiglas using triac-controlled incandescent lamps with networked control, and finally manufactured using aluminum, glass, and neon tubes. We chose the size of the tile to be 66 cm across to match our estimate of average stride length for adults. Fig. 2 shows the inside of the final tile. A neon dimmer controls the brightness of the three neon tubes. A controller board acquires load information from the sensors and communicates with the automation bus.

3.1. Network control of the sensor actuator floor

After a long struggle with our own notions of how the tiles could be networked as cellular automata, we realized that the experts on real-time robust networking for sensor-actuators are the developers of factory automation systems, and we settled on using an established factory automation network called Interbus (www.interbusclub.com). Interbus has been widely used in factories since the late 1980s and has several features that make it suitable for use in Ada's floor. It is a master/slave bus: a single personal computer (PC) with several Interbus master boards can control the entire floor. It is good for automation of devices with

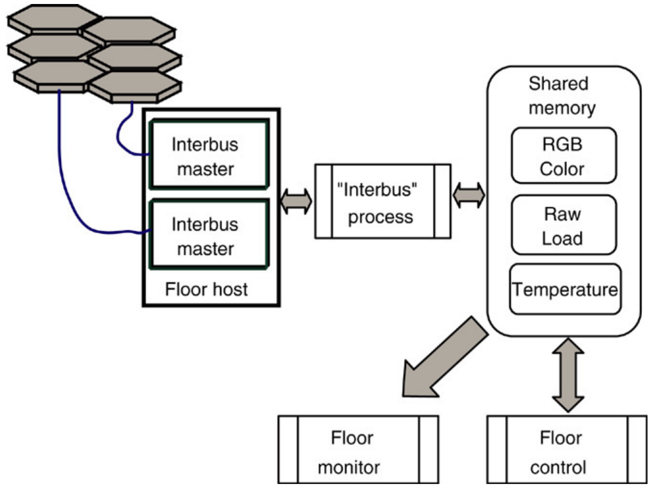


Fig. 3. The interface between the Interbus network controlling and sensing the floor tile states and the host master computer.

a small amount of regularly sampled sensor and control data and has robust error checking and diagnostic capabilities. Floor installation is greatly simplified because the daisy-chained floor tiles automatically number themselves sequentially along the bus. Commodity parts (SUPI3) that implement the protocol are available.

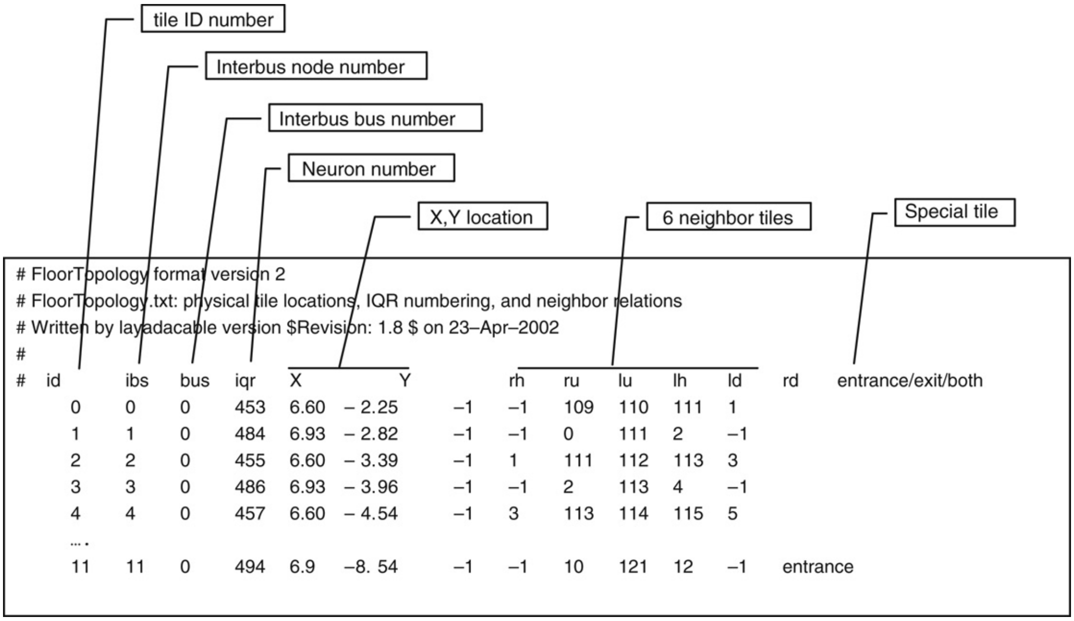
Interbus uses a long data buffer that travels along a 500 kbps differential-signal daisy chained bus. The Interbus protocol hardware on each tile extracts its lamp settings from its portion of the data buffer and replaces it with load sensor data. The cable carries both outbound and inbound connections so that the sensor data can pass back to the master without the last tile on the daisy chain needing to be connected directly to the master; in addition the bus is automatically terminated at the end of the chain.

3.2. *Interbus master interface*

Communication between the higher-level floor-controlling software and the network of floor tiles is via a daemon-like process that uses the Interbus master communication interface driver to provide a shared memory interface to the higher-level software (Fig. 3). This interface allows development of the higher-level functionality to be decoupled from that of the hardware. The floor tiles appear as a set of shared memory segments. An arbitrary number of processes can read from any segment, whereas only a single process can write to a segment. Separate segments represent the loads on the tiles, the colors to be displayed, and the temperatures of the individual tiles. A single Interbus network is limited to a total of about 120 slave nodes. We used four separate Interbus buses to control Ada's 360 tiles, and interfaced to each bus with a separate master controller (Hilscher Automation CIF 50-IBM). All the master controllers were installed in the same PC. Ada's Interbus-based floor ran with an update rate of about 50 Hz at the driver interface level.

3.3. *Toplogical interbus floor configuration*

An Interbus node has no physical location and no 2-dimensional topographical relation to other nodes. The physical topology of the floor needs to be superimposed on the virtual network topology in software. We developed a Matlab script of about 1000 lines to allow drawing of the routing of the Interbus cabling superimposed on the architectural drawings of the floor installation. The Matlab script writes a configuration file that specifies the numbering of the tiles, the physical locations of the tiles, and the neighbourhood relations between the tiles. It also specifies the locations of special entrance and exit tiles. This file is used by the floor software to construct topological and geographical relations between tiles. Part of the floor configuration file is shown in Fig. 4.



```
# FloorTopology format version 2
# FloorTopology.txt: physical tile locations, IQR numbering, and neighbor relations
# Written by layadacable version $Revision: 1.8 $ on 23–Apr–2002
#
#  id     ibs  bus  iqr  X       Y              rh   ru   lu   lh   ld   rd   entrance/exit/both
     0    0    0    453  6.60  – 2.25      –1   –1   109  110  111  1
     1    1    0    484  6.93  – 2.82      –1   –1   0    111  2    –1
     2    2    0    455  6.60  – 3.39      –1   1    111  112  113  3
     3    3    0    486  6.93  – 3.96      –1   –1   2    113  4    –1
     4    4    0    457  6.60  – 4.54      –1   3    113  114  115  5
     ... .
     11   11   0    494  6.9   –8. 54      –1   –1   10   121  12   –1   entrance
```

Fig. 4. A portion of the floor configuration file produced by the topology configuration tool.

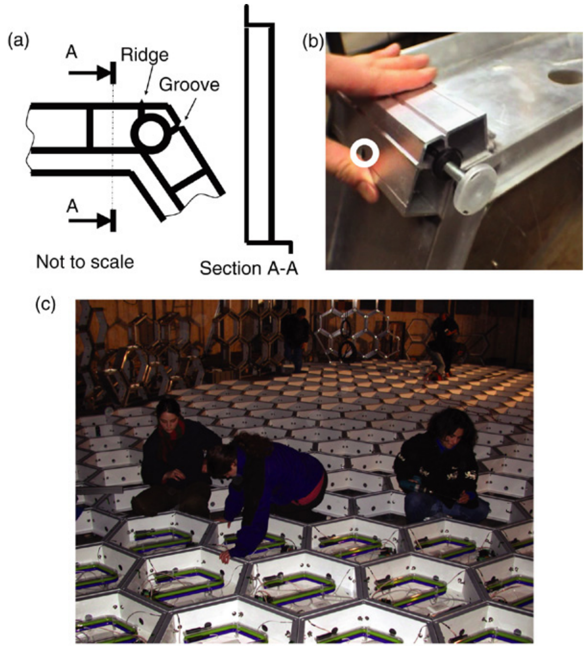


Fig. 5. Mechanical aspects. (a) Tile frame profile shapes (not to scale) and outside corner of frame. Arrows show the mating alignment extrusions. (b) The circle shows one of the holes for bolting the tiles to each other. (c) During installation.

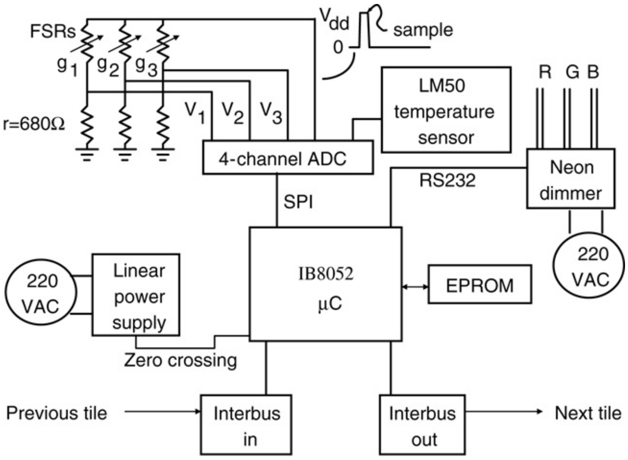
3.4. Neon lighting

The floor tiles were illuminated by neon tubes. Neon lighting is a mature technology that is reliable and very power-efficient and there are many firms with experience in manufacturing it. Light-emitting diode (LED) illumination was also considered, but was dismissed as being too expensive in 2002. Although LEDs are still expensive, it is expected that later versions of the floor will switch to LED lighting.

Three neon tube lamps – red, green, and blue (RGB) – illuminate the tiles and are shaped in the sigmoid form shown in Fig. 2. They are controlled by commercial 3-channel, 80 mA, 990 V neon dimmers (www.toni-maroni.de). Each dimmer cost about USD 200. Power is supplied to the tile as 220 VAC and is locally converted for use by the neon lights and the controller board. Maximum tile power consumption is about 100 W. The maximum tile brightness of 200 cd m^{-2} is comparable to that of a computer monitor display. Measured colour brightness versus lamp setting curves were used to linearize the lamp output.

3.5. Physical construction

The design of the tile frames and tops was important because the floor had to withstand heavy pedestrian traffic and light wheeled vehicles (gantries, equipment trolleys and wheelchairs) over months of operation. We also wanted to be able to transport and reinstall the floor at another location relatively easily with an arbitrary shape. Fig. 5 shows how the tile frames are built from extruded aluminium, using two extruded aluminium shapes cut in sections and welded to form the frame. One piece forms the walls of the frame, and the other forms the legs. Adjustable feet are press-fitted into the legs, and 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. For safety, in the event that water pools beneath the tiles, the electronics inside the tile are mounted above a removable hexagonal plate, which is in turn mounted above the underlying floor level. Two types of special passive tile units fill in the edges of the floor to make them straight.



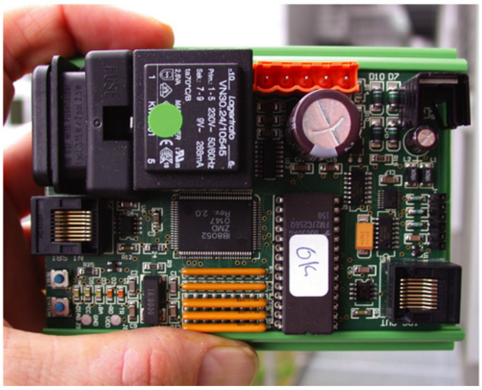


Fig. 6. The floor tile slave controller board.

The tile tops are made from two layers of 8 mm tempered safety glass bonded together by translucent polyvinyl butyral (PVB). An additional 3 mm translucent polycarbonate plate under the glass further diffuses the light to better mix the colours. Each glass top costs about USD 100.

Translucent plastic tops were considered because they would have been relatively inexpensive and would have better mixed the colors. However, Plexiglas tops were rejected by the exhibition safety officials because acrylic plastic emits toxic fumes in a fire, and polycarbonate tops of the required thickness were not available. In any case, plastic tops produce very unpleasant static discharge, and would have become easily scratched and difficult to clean.

3.6. Floor tile slave controller

A local slave controller in each tile (Fig. 6) reads the sensors, controls the neon dimmer to set the lamp RGB brightness, and communicates with the Interbus. It also enables self-diagnostics for the lamps and load sensors and has an automatic sleep mode that turns off the lamps and reduces the sensor sampling rate after a period of inactivity. Zero-crossing detection of the power line cycle is used to synchronize lamp brightness changes and sensor acquisition to the line cycle to reduce lamp flicker and sensor noise. The tile controller uses a MAZeT IB8052 microcontroller with embedded Interbus link level controller logic (New designs should use a microcontroller that can communicate with an INTERBUS SUPI3 interface chip). The firmware is about 800 lines of C code. Each tile controller board cost about USD 70.

Power is routed through the floor and to the tiles using commercially available 3-way IEC equipment cables. Approximately 20 tiles can be powered through each such daisy-chained arrangement. Power to the entire floor is thus typically supplied from one edge, with higher-capacity cables supplying sets of rows of tiles.

3.7. *Load sensors*

The public visiting Ada ranged from young children weighing about 20 kg to adults weighing over 150 kg. Large adults jumping about can transiently weigh several hundred kilograms. The most important requirement to support accurate people tracking is to reliably detect loaded tiles without falsely detecting unloaded tiles as loaded. This seemingly simple task proved to be challenging in the context of a large, long-running public exhibit.

We used force-sensitive resistors (FSRs) as our load sensors after considering several alternative technologies. FSRs are flat and robust, although poorly matched. We mounted three of these, equally spacing them at three corners of the hexagonal tile frame between the frame and the glass tile top and under a 3 mm layer of grey adhesive-backed ethylene propylene diene monomer (EPDM) rubber. This ring of rubber supports the glass and is also important because it divides the force seen by the rather sensitive FSRs to a usable range for human weights. We chose EPDM rubber because only natural rubber has higher resilience but EPDM loses its resilience much slower. However, it does stay compressed after sustained heavy loading and can take hours or days to recover its uncompressed form.

FSR conductance increases monotonically with the applied load, approximately as a square-root relationship. FSRs are effectively single-sourced (www.interlinkelec.com; another source recently discovered is www.tekscan.com) and cost about USD 5 each. Although three sensors are sufficient for sensing the load on a single tile, six would have ensured that there were no blind spots on the assembled floor and would have resulted in more reliable tracking of visitors who occasionally stand on tile intersections where there are blind spots in the load sensing. (Most intersections are covered by one or two load sensors, but owing to the pseudo-random assembly of tiles and floor, there are some intersections with no load sensor.)

The tile measures the load on its glass top by forming a voltage divider with each of the three FSRs (Fig. 6). FSRs can be damaged by sustained continuous current, so they are only powered during readout, with an active duty cycle of less than 5%. The sampled ADC readings V_k from the FSR voltage dividers are used to compute the sum of the three normalized FSR conductance values rg_k to form a single raw tile load value G which is linearly related to the load applied to the tile in the range the FSRs are used, as shown in (1):

$$\begin{aligned} &g_k\,(V_{dd}-V_k)=V_k/r \\ &rg_k=\frac{V_k}{V_{dd}-V_k} \\ &G=\sum_{k=1,2,3} rg_k. \end{aligned} \tag{1}$$

3.8. *Load signal processing*

The tile-to-tile variation in G is about one-third of the observed full-scale value. Manufacturing differences in the tile frames and tops cause most of this huge mismatch by applying varying amounts of the load from the glass to the frame instead of to the FSRs. Moreover, the excitation of the neon tubes by 1 kV, 20 kHz voltage pulses causes a significant impulsive noise spike if a neon excitation pulse occurs while a nearby load sensor is being read.

Active visitors shift the glass in the tile frame, resulting in significant shifts in the baseline tile load. As the tile rubber between the FSRs and the glass ages or becomes compressed by sustained loading – which occurs, for instance, near an exhibition entrance – the baseline tile loads slowly change.

This impulsive noise and non-stationary tile–load–sensor mismatch requires filtering the raw load signals before determining whether tiles are loaded. Filtering is computed on the floor host controller but is independent for each tile.

The impulsive noise is first removed by applying a running median filter to G with a window of five to nine samples, resulting in the median-filtered raw load values R. This median filter is very effective but adds a latency of about half the window length for changes in load.

The floor controller then continuously estimates the unloaded state U of each tile. U is subtracted from R to produce the final filtered load L and a fixed positive threshold T then determines the binary loading state B,as shown in (2):

$$\begin{aligned} &L=R-U \\ &B=\begin{cases}1 & \text{if } L\geq T\\ 0 & \text{if } L<T\end{cases} \\ &U=\ \text{filtered } R \text{ value.} \end{aligned} \tag{2}$$

This filtering allows a single global T to be set so that a 5 kg force applied by a few fingers pushing down anywhere on any of the 360 tiles (except directly over one of the corners without a load sensor) is enough to trigger a loaded state.

U is updated by linearly slewing its value towards R using asymmetrical loaded and unloaded slew rates S_{loaded} and S_{unloaded} and the time step Δt:

$$U_{t+\Delta t}=U_t\,\Delta t\times\begin{cases}-S_{\text{loaded}} & \text{if } L>0\\ S_{\text{unloaded}} & \text{if } L<0.\end{cases} \tag{3}$$

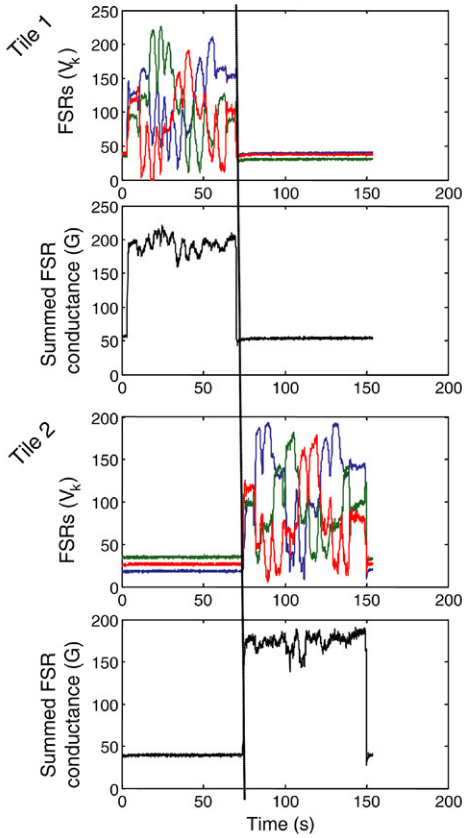


Fig. 7. Shows data from the load sensors from two tiles while a person moves to various locations on the tiles, along with the summed FSR conductances. At time 75 s, the person steps from Tile 1 to Tile 2.

S_{loaded} is $1/300\ \text{s}^{-1}$ to slew full scale, S_{unloaded} is $1/10\ \text{s}^{-1}$ for full scale. This asymmetry is necessary so that tiles can become resensitized rapidly to small loads (active children) while only slowly forgetting about large loads (adults lingering in one place). Thus U represents a kind of "dip detector" that represents the average unloaded tile state. These learned U states are saved persistently to enable rapid startup. They constitute one of the forms of long-term memory in Ada. *This continuous adaptation was essential for ensuring reliable visitor interaction*.

The raw sensor values and combined raw tile load values G from two tiles are shown in Fig. 7. A person stepped from Tile 1 to Tile 2, and, while on each tile, moved around to different locations on the tile. Although the individual sensor values fluctuate considerably depending on where the feet are located, the combined values remain relatively constant, indicating that the combined value is useful for detecting a loaded tile. The remaining variability in the load value results from the person's movement and the use of three rather than six FSRs.

Fig. 8 shows measurements of tile load sensor linearity from the same two tiles using a range of three loads generated by two people standing either alone or together on the tiles. Although the tile DC values and gains are different, each tile's response is linear in the applied load.

Although reliable binary detection of loaded tiles is the most important use of the load information, Ada also uses L to estimate the height of people's heads to better aim its gazers to look at them, and in one of the games described later (Boogie) to measure the rhythmicity, synchronization and power in visitor dance movements.

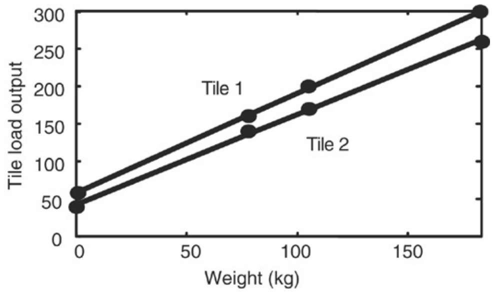


Fig. 8. Tile load linearity. The raw load values G for two tiles are shown.

4. Floor control software

Ada's software as a whole is a heterogeneous mixture of procedural code and neural networks [11,13]. The floor software is entirely procedural because our timeline did not allow for research into controllably embedding complex dynamical behaviors (e.g. games) in neural networks. All of the floor code except for the shared memory Interbus interface is written in Java. We chose Java for its high productivity, debugging tools, and remote procedure call support. Hard real-time performance was not critical for this exhibit, but we were pleased by Java's ability to deal with a large, soft real-time environment.

A single 1 GHz PC handles the entire floor. It runs the C++ Interbus process (13 classes, ∼6500 lines of code) and a Java (1.4 JVM) floor process comprising about 80 Java classes with about 20,000 non-comment lines of code. The floor processing occurred in a single main thread that cycled over (1) tile load acquisition and signal processing, (2) state-specific processing, and (3) tile rendering cycles. Particular behaviours plug into this framework using a uniform interface. Other machines on the network access the floor using remote method invocation or shared memory "neural activity maps". Ada's exhibition-floor update rate running on a 1 GHz Pentium III processor under a load of about 20% was about 15 Hz, fluctuating due to time slicing, garbage collection, network congestion and other factors. Occasional pauses of up to a half second were sometimes seen and were thought to be due to delays in software components running on other machines. This irregularity was acceptable for an exhibition. Present improvements in processor speed and Java virtual machine technology would increase update rate significantly.

4.1. Rendering

Although an interactive floor might seem to be like a large, low-resolution version of a touch-screen, traditional two-dimensional image rendering is not appropriate for controlling the colours of Ada's tiles because large interactive floors are inherently multi-user, requiring multiple localized input–output interactions, and the tiles near any one visitor dominate that person's view. Therefore we developed classes that render fluid localized patterns of activity which could be linked to a tile or to a tracked visitor. For example, tiles can be lit, pulsated or flashed. Pulsating, expanding and contracting rings and blobs can be displayed and directional effects can be generated. We also developed a general set of dynamic patterns that can be displayed on floor regions, such as pulsating or cycling floor colors, drifting sinusoidal gratings and perimeters of floor regions. Together, these local and global effects provide the outputs onto the floor. Reactive effects that are automatically created on freshly loaded tiles constitute most of the behaviour of Ada during its simpler behavioural modes like sleeping or encouraging visitors to leave.

Created visual effects are placed on a list and update themselves at every floor cycle until they expire or are removed. Each effect object knows how to update its dynamic state, how to compute which tiles should be affected, how to set their brightness, and so forth. A large number of general parameters such as color, transparency, rate, size, rise, and fall time control the appearance and dynamics of the effects.

4.2. People tracking

The primary objective of Ada is to identify individuals and playfully interact with them. To enable complex interactions, Ada tracks visitors so that labels assigned to them – a special colour or pattern – can travel along with them or a gazer or light finger can follow them. The result of the tracking algorithm is a list of tracked people maintained by the floor server process. Each 'person' object carries information useful for Ada's behavior: its tile location, the estimated average tile load, visual effects assigned to that person, gestures (described later), and the person's average direction of motion. In addition, global statistics such as the fraction of loaded tiles and the average rate of tile crossings were maintained and used as inputs to the overall Ada state controller. The list of tracked people and global statistical information are available to other processes or machines by shared-memory and remote method calls.

The real time matching-based tracking algorithm (Fig. 9) is applied during each load sensor update cycle. The tracker is based on a heuristic model of a person as an object that occupies an area of a single tile and that can only move to nearby tiles. The filtered weight sensor values (as described earlier in Section 3.8) determine whether a tile is loaded or not and tracking is based on these binary loaded tile states. Tracked persons are assigned to a single tile.

Tracking starts when a tile not belonging to a tracked person is loaded for at least 300 ms and is surrounded by unloaded tiles (Fig. 9(a)); the delay prevents initiating tracking on transiently loaded tiles. When a tile assigned to a tracked person becomes unloaded, nearby (within two tiles) loaded tiles that have not been loaded for too long (typically 700 ms) are assigned as possible destinations of that person (Fig. 9(b)). Tiles that have been loaded for too long are considered to belong to other visitors who may not be tracked yet or who are standing on several tiles. A list is built of all possible destinations of all tracked persons. This destination list includes the source tiles if they are loaded. This list is pruned by matching person to destination. As each match between target and destination is made, the corresponding objects are removed from further consideration using the rules and priorities illustrated in Fig. 9(c).

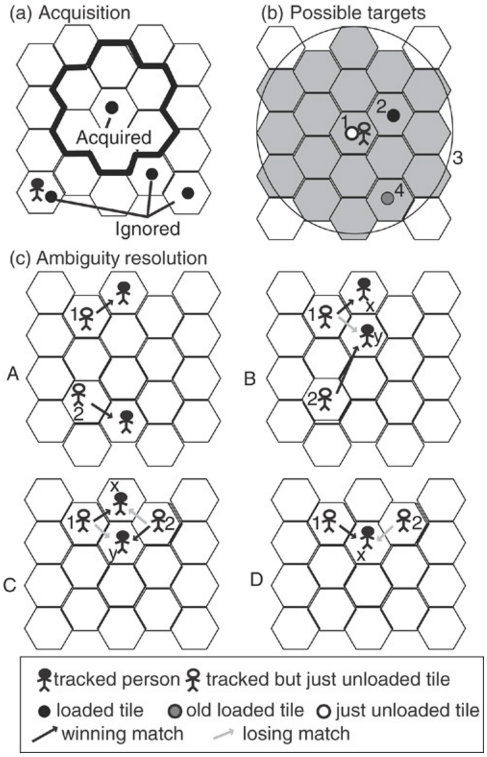


Fig. 9. Person tracking algorithm. (a) Persons are only acquired for tracking when they occupy a single isolated tile. The isolated tile starts a person, but the other loaded tiles do not because they either belong to a tracked person or are not isolated. (b) Finding possible matches for a person. A person at 1 steps off the tile. Loaded tiles within search region 3 are added to the list of possible destinations. Tile 2 is added, but tile 4 is not added, because it became loaded too long before tile 1 was unloaded. (c) Sources and destinations are resolved in the order A-D. In A there is no ambiguity; the destinations of persons 1 and 2 are clear because they are the closest possible matches. In B, 1 has ambiguous destinations *x* and *y*, but 2 does not; these are resolved as 1 goes to *x* and then 2 goes to *y*. In C, the targets of 1 and 2 are both ambiguous; the destinations are arbitrarily chosen as shown. In D, the only destination for both 1 and 2 is *x* and one person is arbitrarily discarded.

If no match is immediately found for a tracked person when the tile becomes unloaded, a timeout is started because the visitor could be jumping. If a proper match is found within the timeout (typically set to 400 ms), then tracking continues; otherwise, the tracked person is discarded. Such situations also allow tracked visitors to generate two kinds of gestures, a *hop* and a *pogo*. A hop is a jump through the air to another tile, while a pogo is a jump in the air that lands on the same tile. These gestures are used in games like Gunfight, which is briefly described in Section 5.

Clearly, there are situations when tracking fails—for instance, when two tracked people come together to stand on a single tile. Then one of them is discarded. But for the most part these situations do not occur, because people (at least adults) maintain a significant personal space.

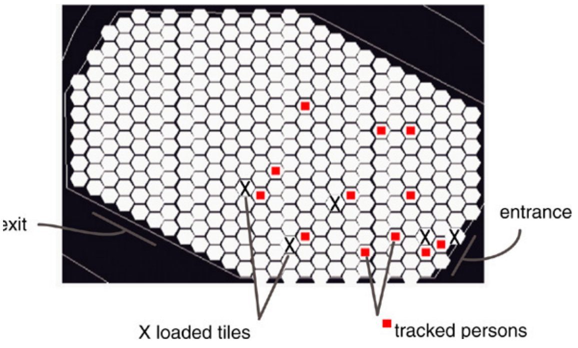


Fig. 10. A snapshot of tracking soon after a group of visitors has entered the space.

The tracker easily runs in real time. Although the final tracker code is only about 500 lines of code, development of a correct tracking algorithm was quite difficult. To debug the tracker, it was necessary to develop a graphical software interface that allowed slow motion manipulation of the loading state of the tiles.

Fig. 10 shows a snapshot top view of the floor during tracking. Tiles labeled with black crosses were loaded and those labelled with red dots were loaded and linked to tracked persons. Most loaded tiles are associated with tracked persons.

We cannot provide a single quantitative measure for the reliability of tracking because tracking reliability depends strongly on operating conditions, which were extremely variable during the exhibition. The key reason that our heuristic matching-based tracking algorithm does a credible job in tracking visitors is because *at the moment that tracked people unload their assigned tile, they have a very limited number of possible destinations*. The tracker performs well when the space is uncrowded (<5% loaded tiles) or when visitors want to interact with Ada. Such visitors usually step on single tiles and keep a polite tile distance from each other. When the space is uncrowded, even completely naive visitors are tracked reliably over distances of many tens of tiles regardless of whether they step on tiles or tile intersections. When the space is crowded, a knowledgeable visitor familiar with tracking, or a naive visitor who steps only on single tiles that are unoccupied, can also be tracked with high reliability. Tracking is significantly degraded in situations where children run wildly through the space trying to step on as many other people's tiles as possible, or by very crowded conditions, when people tend to ignore tile boundaries. We show some representative data in the form of recorded long tracker paths in Fig. 11. Baebler's analysis [15] of correlated floor-tracking data and recorded video showed that most tracking errors were related to unreliable detection of loaded tiles – probably due to rubber compression and the use of three rather than six FSRs per tile – rather than to incorrect matches by the tracking algorithm.

5. Games

High-level reactive and interactive behaviours and the Football game were described in the Section 2. We also developed a number of other games. Complex games were far less effective in engaging the public than simple games that can be learned by watching for a few seconds.

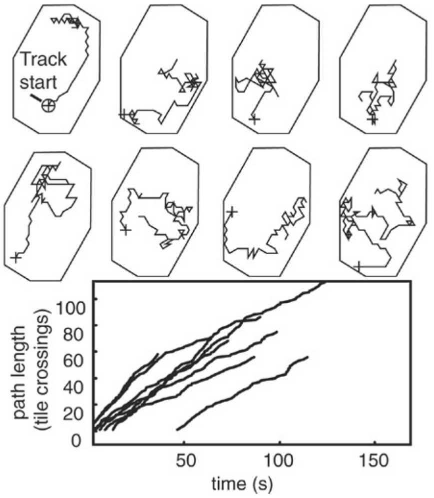


Fig. 11. Representative tracker paths from the longest 10% of all paths, over 2 min period on 19 October 2002, along with tile crossing data vs. time.

These games generally consist of a few hundred lines of code because they make extensive use of the underlying support classes as described in Section 4. Games are controlled by a state machine that updates itself on each floor update cycle. Games also implement a uniform interface that allows their uniform integration into the floor processing cycle, and are started or stopped based on an overlying neural network-based state machine whose transitions are influenced by variables such as occupation density and time of play [11].

"*Pong*" splits the floor into two halves, with a virtual paddle on each half of the floor which is collectively controlled by the median location of the players on that half of the floor, leading to spontaneous cooperation among strangers. Teams seek to shoot the ball into the opposing team's goal.

"*Boogie*" is a collective dance game or prototype automatic disco. The power spectra of the analogue load information from active dancers are analysed to extract dominant frequencies, and a consensus drives the overall rhythm and volume of the dance mix. Inactive participants who simply stand or walk about are ignored, so a few active dancers can easily dominate the rhythm of the entire space.

"*Gunfight*" labels tracked players with lit tiles; players use "hop" gestures – where they jump to an adjacent tile – to shoot virtual bullets toward other players to extinguish them from the game. Players can use "pogo" gestures – where they jump up and land on the same tile – to temporarily surround themselves with an impenetrable shield. The surviving victor is rewarded with impressive collapsing rings of green covering the entire floor.

"*HotLava*" was inspired by a television game show. Two players compete to find their way across the floor on a hidden path. If they step off the path, their half of the floor flashes an angry red, and they must start over from the beginning, while trying to remember their previous steps.

Finally, in a simple but effective game called "*Squash*", each tile on the floor is randomly illuminated with one of two colours. Two teams are formed, and each team is assigned a colour. The teams then compete to see who can first extinguish all the tiles of their own team's colour by stepping on them. (Stepping on tiles of the opponent's colour only wastes time and helps the opposition.) This game is good for a battle of the sexes.

6. Discussion

We believe that this is the first instance of a manufacturable combined luminous and tactile floor. In the course of building the floor over three years and three major prototypes we developed novel techniques for analogue load sensing, signal processing of load information, tracking of human users and rendering visual effects. The floor is also a novel use of a factory networking technology, and we invented floor-based games and interactions that could be learned and enjoyed in a few moments. It is further distinguished by its size and toughness: the 360 active tiles covered 136 m^2 and were used for five months, seven days a week, 12 h a day, receiving over 550,000 visitors during that time. The uptime of the tile electronics was well over 99.5%. The only mechanical problems found were some ageing of the tile rubber in individual cases.

This key component of Ada (our playful interactive space) could have general usefulness in other autonomous interactive robotic environments. This tactile luminous surface offers new forms of human–machine interaction, and the Ada exhibit demonstrated that it could be used reliably with the general public. Ada's floor was the key component that saw it voted by the public as one of the best five out of 60 exhibits at the USD 600M Swiss National Exposition of 2002. Such expositions only takes place once every 30–40 years. The high cost (USD 800/tile) of the custom-produced present floor limits its applications, but this cost would come down significantly in higher-volume production outside Switzerland, especially as the price of high power LEDs drops.

Acknowledgments

Ada (www.ada-exhibition.ch) was funded by the University and ETH Zürich, Manor AG, and the Velux and Gebert Rüf foundations. G. Ritler constructed the floor tile bases for the prototype floors. Westiform AG's design, management, and construction (E. Lüdi, M. Gloor, and H. Peyer) enabled the final form of the floor. O. Hasselbusch (OHTec) assisted in the design of the production control board. We thank U. Rutishauser, M. Erzinger (ETH Zürich), and R. Derrer for valuable technical and management assistance.

References

[1] E. Johnstone, A MIDI foot controller — The PodoBoard, in: 1991 International Computer Music Conference, San Francisco, 1991, pp. 123–126.

[2] R. Pinkston, et al., A touch sensitive dance floor/MIDI controller, in: 1995 International Computer Music Conference, San Francisco, 1995, pp. 224–225.

[3] J. Paradiso, et al., The magic carpet: Physical sensing for immersive environments, in: CHI '97, Atlanta, Georgia, 1997, pp. 277–278.

[4] N. Griffith, M. Fernstrom, Litefoot — A floor space for recording dance and controlling media, in: 1998 International Computer Music Conference, Ann Arbor, Michigan, 1998, pp. 475–481.

[5] T. Morishita, T. Sato, Robotic room: Its concept and realization, Robotics and Automation 16 (1998) 705–717.

[6] R.J. Orr, G.D. Abowd, The smart floor: A mechanism for natural user identification and tracking, in: Computer Human Interaction, New York, 2000, pp. 275–276.

[7] H. Morishita, et al. 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, 2002, pp. 1246–1251.

[8] J. Paradiso, et al., Sensor systems for interactive surfaces, IBM Systems Journal 39 (2000) 892–914.

[9] K. Eng, et al., Ada: Constructing a synthetic organism, in: 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2002, EPFL, Lausanne, Switzerland, 2002, vol. 2, pp. 1808–1813.

[10] T. Delbruck, et al., Ada: A playful interactive space, in: Interact '03 IFIP TC13 International Conference on Human–Computer Interaction, Zurich, 2003, pp. 989–992.

[11] K. Eng, et al., Design for a brain revisited: The neuromorphic design and functionality of the interactive space 'Ada', Reviews in the Neurosciences 14 (2003) 145–180.

[12] K. Wassermann, et al., Live soundscape composition based on synthetic emotions: Using music to communicate between an interactive exhibition and its visitors, IEEE Multimedia 10 (2003) 82–90.

[13] K. Eng, et al., A space that learns to influence human behaviour, IEEE Transactions on Systems Man and Cybernetics, Part A 35 (2005) 66–77.

[14] T. Delbruck, et al., Device for controlling a physical system, US 6,603,082, EP1091334, JP2001180810, issued 2003, 2004.

[15] A. Baebler, Multimodal people tracking in a cluttered environment, Ph.D. Thesis, ETH Zurich & Univ. of Zurich, Zurich, 2004.

Tobi Delbrück is a group leader in the Institute of Neuroinformatics (INI). His main research interest is the development of useful neuromorphic devices, particularly vision sensor chips. He studied physics and applied mathematics as an undergraduate and received his Ph.D. from Caltech in 1993 in the department of Computation and Neural Systems. He then worked for four years as a consultant in Silicon Valley on electronic imaging with Synaptics, National Semiconductor, and Foveon and moved to the INI in 1998. He has been awarded eight patents, and has over 30 refereed papers in journals and conferences, four book chapters, and one book. He has been awarded four IEEE awards, including the 2006 ISSCC Jan Van Vessem Outstanding European Paper Award. He is a visiting scientist at Caltech. His responsibilities in the Ada project were the hardware and software development of the floor, including the tile control board and its firmware, the cooperation between the INI and Westiform, the software architecture of the floor software, the physics of the load sensors, the load sensor signal processing, the generation of visual effects, the later evolution of the people tracking, and the development of the floor-based reactions, interactions and games.

Adrian M. Whatley gained a B.Sc. in chemistry at the University of Bristol in England in 1986. After working for almost 10 years in the British computer industry, he took up his current software engineering position at the INI where he works primarily on Address-Event communication systems. His responsibilities in the Ada project included developing and maintaining the build environment for the entirety of the Ada software, the hardware and software interface between the Interbus network and the Java floor host controller, including the C++ Interbus floor control process, and the management of the on-site floor assembly.

Rodney Douglas is Professor of Neuroinformatics, and Co-Director at the INI. He graduated in science and medicine, and obtained a doctorate in neuroscience at the University of Cape Town. Thereafter he joined the Anatomical Neuropharmacology Unit in Oxford, where he continued his research on the anatomy and biophysics of the microcircuitry of cerebral cortex together with Kevan Martin. As Visiting Associate, and then Visiting Professor at Caltech, he extended his research interests in neuronal computation to the modelling of cortical circuits using digital methods (together with Christof Koch), and also by the fabrication of analogue VLSI circuits (together with Misha Mahowald). In 1996 he and Kevan Martin moved to Zurich to establish the Institute of Neuroinformatics. In 2000, he was awarded the Koerber Foundation Prize for European Science. His responsibilities in the Ada project included overall guidance of the project, the software architecture for the multi-host integration of the components of Ada, the initial development of the people tracking code for the floor, the design of key behaviours of Ada, the overall design of Ada's object-oriented software architecture and the remote interfaces between the floor and other Ada components.

Kynan Eng received degrees in science (applied mathematics/computer science) and engineering (mechanical) from Monash University, Melbourne, Australia, and his Ph.D. from ETH Zurich. His research interests include human-centred interactive environments and robotics. After a number of years in industry working on telecommunications fault management software and gas turbine design for power generation, he joined the INI in 2000 to work on the Ada project. His main responsibilities within the project were the overall technical system integration, development of the behavioral control strategies, on-site operation of the exhibit and the design and execution of human–machine interaction experiments.

Klaus Hepp, born 1936 in Kiel, studied mathematics and physics at the University of Münster and ETH Zurich. He received his Ph.D. in 1962. From 1964 to 1966 he was a member of the Institute for Advanced Study in Princeton. Since 1966 he has been professor of theoretical physics at the ETH Zurich. Hepp made important contributions to quantum field theory and quantum optics, as well as to the neurobiology of eye movements. In 2004 he received the Max Planck Medal of the German Physical Society. He works today as professor emeritus at the INI. In the Ada project his responsibilities included providing project guidance, assistance in constructing the first tile prototype, Ada's gazers (with A. Baebler) and on analysis of recorded data.

Paul F.M.J. Verschure (1962) was a group leader at the INI and is now at the university of Barcelona. He received his masters and Ph.D. in psychology. His scientific aim is to find a unified theory of mind, brain and body through the use of synthetic methods and to apply such a theory to the development of novel cognitive technologies. He has pursued his research at different institutes in the US (Neurosciences Institute and The Salk Institute, both in San Diego) and Europe (University of Amsterdam). He works on biologically constrained models of perception, learning, behaviour and problem solving that are applied to wheeled and flying robots, interactive spaces and avatars. In addition to his basic research, he applies concepts and methods from the study of natural cognition to the development of interactive creative installations and intelligent immersive spaces. Since 1998, he and his collaborators have built a series of 17 public exhibits of which the most ambitious was the exhibit Ada. Verschure leads a multidisciplinary group of 15 doctoral and post-doctoral researchers that include physicists, psychologists, biologists, engineers and computer scientists. His responsibilities in the Ada project included the overall management of the project, the neural simulation software used for parts of Ada, the design and implementation of Ada's overall behavioural state controller, scenography and user interaction studies.