Sensate media — multimodal electronic skins as dense sensor networks

J A Paradiso, J Lifton and M Broxton

In this paper, we introduce the concept of building electronic sensate skins as extremely dense, multimodal, systolic sensor networks. In this fashion, the copious signals produced by the skin's receptors are reduced by the network itself, and only high-level features are routed out peer-to-peer, avoiding complex wiring requirements while promising to enable scalability across large areas. Our architectures and algorithms have been inspired by biological skin, where signals from receptors are enhanced or suppressed by processing in the receptor cells and nervous system before arriving at the brain. We illustrate this concept with work in progress on two test beds, the Pushpin Computer, an easily configurable, planar array of over 100 nodes set up as 'smart wallpaper', and the Tribble, a sphere coated with over 500 diverse tactile and noncontact sensor channels processed in 32 interconnected sensor 'cells'. We also introduce the Z-Tiles, a collaborative project that has resulted in a pressure-imaging floor realised through a collection of pixilated floor tiles configured as a mesh sensor network. We conclude with a technology roadmap for scaling to higher densities.

1. Introduction — towards sensate media

Biological skin, with its huge density of sensor pick-ups (e.g. 250 receptors/cm^2 in the human fingertip) and multimodal sensor capability [1] (vibration, pressure, shear, tactile flow, temperature, pain, even optical in the case of animals like the brittlestar [2] or capacitive proximity for electric fish [3]), has no parallel in the world of today's electronics. Were such electronic skins to be developed, they would precipitate revolutions in fields such as robotics (where contact sensors contribute a rich sense of tactile presence, and proximity sensors provide warning before collisions, a vital element of physical embodiment [4]), prosthetics (artificial limbs that approach the sensory capability of their biological counterparts), telepresence (providing rich immersion), and telemedicine (giving a doctor or surgeon detailed tactile feedback during a remote procedure or examination).The benefits and technological challenges of developing skins of this sort are elucidated in Lumelsky et al [4, 5], a call-to-arms to the sensor technology community to develop flexible, multimodal, sensate membranes that can meet or exceed the sensing capabilities of human skin. Lumelsky et al [5] strongly assert that the development of such electronic skin will spur theoretical advancements in a wide variety of fields, producing a research stimulus that '... will be comparable to that which triggered the explosion of control theory in the 1940s and 1950s, in direct response to the challenge posed by the appearance of fundamentally new hardware, such as jet fighters and radars'.

Many researchers have developed different types of tactile sensors for cybernetic applications that exploit a wide variety of technologies [6]. Examples include densely printed passive force-sensitive resistor (FSR) arrays for commercial pressure imaging applications [7], active FSR arrays for robotic hands [8], shear-sensing laminates for aircraft wings [9] (the term 'smart skin' is well over a decade old in aerospace research [10]), capacitive matrices for human-computer interface (HCI) applications [11—15], woven, fabric-based capacitive and resistive pressure and strain sensors [16], and large-area arrays of diverse sensors fabricated on flexible substrates [17, 18]. Looking at examples more explicitly addressing cybernetic applications, NASA's Robonaut project employs many sensors on a telepresence robot to provide feedback to the operator [19], Stiehl and collaborators at the MIT Media Lab have tiled a robotic hand with an array of 42 force-sensitive resistors [20], and on a smaller physical scale, there are a variety of efforts concerning artificial finger tips [21—23].

our architectures and algorithms have been inspired by biological skin

Essentially all signals from patches of today's ultradense electronic skins are wired out directly from each receptor (or common row/column) in unprocessed analogue form. Most of these constructs tend to involve a maze of complex wiring, with every signal routed to multiplexing arrays and a central digital processor, as depicted in Fig 1. Although some techniques are wireless (e.g. those using optical imaging of a deformable dot matrix on an elastomer membrane [21, 24],

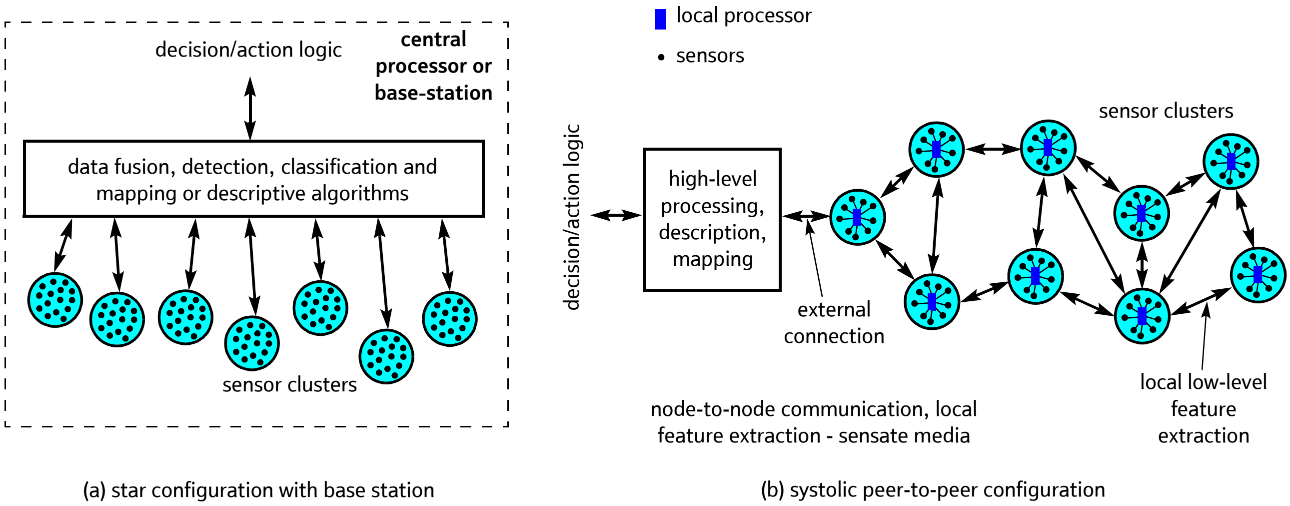


Fig 1 Standard, multiplexed sensor architecture (left), with all nodes (or analogue sensor signals) communicating with a heavy base station and sensate media architecture (right), with smart nodes that collaboratively reduce data into higher-level features at the point of collection, which are then externally routed as needed.

those powering wireless sensor tags embedded into a skin with an AC magnetic field akin to RFID [25], and those powering skin-embedded sensor tags optically [26]), their communications range is quite limited, requiring many readers to be embedded into a potentially bulky substrate. A practical skin requires a distributed set of networked data concentrators to handle discreet patches of sensor inputs. Examples of this approach can be seen in the modular array of IR proximity sensors developed at the University of Wisconson for electronic skin on a robot arm [27] and the interconnected piezoelectric impact-sensing units for the smart spacecraft skin of the 'ageless aerospace vehicles' concept demonstrator at CSIRO in Australia [28].

The amount of data that even a modestly sized piece of such a skin can generate, however, can be enormous. Although different receptor families can be interpreted to be sampled at varying rates and (potentially nonlinear) resolutions [29—31], an approximate biological analogue would involve 8-bit digitised samples taken at 500 Hz, yielding a 12.5 Mbyte/s stream for a 10×10 cm patch of fingertip-density electronic skin. In order to make such skins feasibly scalable, this large data stream needs to be reduced in the skin itself. We see something of this principle in biology [32], where signals from skin receptors are affected by interaction in the nervous system on the way to the somatosensory cortex in the brain [29, 33]. Sensory receptors dynamically adapt to static stimuli and resulting nerve signals are spatially enhanced or suppressed via lateral inhibition as nerve fibres join at synaptic junctions in the spinal column and thalamus. Since waiting for a signal from the brain can be too slow to avert an immediate physical crisis, muscles are directly controlled by autonomous spinal reflexes in the event of a sudden painful stimulus like a burn or sharp stick.

Our work views such dense electronic skins as sensor networks, with a small, embedded processor dedicated to a local cluster of sensors. It is well known that maximum throughput of independent data in a mesh network will scale with the square root of the number of nodes (n), thus the bandwidth available per node scales as $1/\sqrt{n}$, which approaches zero as $n \to \infty$ [34]. As such, in order to maintain global bandwidth across a large sensor network, features must be locally extracted and the data accordingly compressed [35, 36] before routing out higher-level results. Obtaining information from all sensors at the pixel level can jam the network and will not allow densely sampled skin to scale to any significant size or bandwidth. In our designs, all processors can communicate with others in their neighbourhood, allowing data from stimuli to be reduced and processed locally across their physical footprint, and resultant descriptive parameters to be routed to external connections, passing from processor to processor. If a node fails, data can be alternately routed, or in a wired network, a noisy node can be encapsulated and effectively removed from the system, leading to a high degree of fault tolerance. We term such sensor-network-based smart skins, as depicted in Fig 1b, to be sensate media.

the amount of data that a skin can generate, however, can be enormous

Research in the sensor network field (e.g. Gharavi and Kumar [37]) tends to look at systems and applications that anticipate nodes distributed on the scale of smart buildings, wildlife preserves, or battlefields, where node spacings range from several metres to kilometres. Although sensate media will exhibit much denser node spacings, from a millimetre to a centimetre, many aspects of its algorithmic and information processing challenges (e.g. data reduction, stimuli characterisation, and feature extraction) are analogous to those faced with larger-scale sensor networks. Related concepts have appeared before in various incarnations, but their focus differed from that of skins (e.g. the Smart Matter project at Xerox PARC [38] concentrated on distributed control, while the Amorphous Computing [39] and Paintable Computing [40] projects dealt mainly with massively distributed processing and most results were gleaned via simulation).

To explore very dense sensor networks that point towards this frontier of sensate media, we have developed two test beds, which we describe in this paper. The first exploits wireless IR communication between nodes that can be arbitrarily distributed atop a flat plane, and the second is a wired mesh sensor network that totally envelopes a 33-cm diameter sphere in an electronic skin. We also introduce a collaborative project that has resulted in pressure-sensitive tiles that snap together to form a sensor network skin that can be placed atop a floor to characterise and track footsteps. We conclude with an examination of issues to be tackled in moving sensate media forward — addressing fabrication, processing, and power challenges to achieve greater densities.

2. Pushpin computing — a planar test bed for high-density wireless sensor nets

In order to effectively penetrate applications and make sensor nets a practical option beyond the research community, advances in resource-constrained distributed computing are required that balance and conserve the consumption of power, processing, and bandwidth. Most development in this area has occurred with network simulation programs that emulate large numbers of interconnected processing and sensing nodes [40—43]. Although simulations have lent some solid guidance and stimulated excellent work, they are unable to properly address the unpredictability of real-world applications, where effects from communications interference and crosstalk, RF noise, and sensor responses to real-world stimuli and background, can be extremely significant.

In order to gain practical experience, various researchers have deployed test beds, typically involving dozens of sensor nodes that communicate with other nodes in their neighbourhood via radio (e.g. Fang et al [44]). When stretching the node count into the hundreds or higher, such test beds can become untenable. As radio communication becomes difficult to reliably limit to distances below 10 metres, these sensor networks become physically very large, making it difficult to apply deterministic stimuli and causing various logistical problems (maintaining, servicing, or updating the nodes becomes time consuming, to say nothing of changing batteries where required) that preclude an agile test bed.

Accordingly, we have built a system that avoids these pitfalls. Termed ‘Pushpin Computing’, the sensor nodes have the form factor of a Pushpin that can be inserted anywhere into a large planar substrate, from which they draw their power (Fig 2). Repositioning a node is a trivial matter, no more difficult than shifting a thumbtack on a soft corkboard, hence the nodes can be readily configured into any planar distribution (Fig 3). Pushpin computing, as introduced in Lifton et al [45, 46], began as an initiative to explore Butera’s paintable computing paradigm [40] in hardware, then evolved as a general tool to explore very dense, distributed sensor networks. The Pushpin research is driven by the vision of sensor networks small enough and dense enough to enable coating surfaces in an electronic skin, as elucidated in the prior section. That said, the Pushpin network might also be considered a model of

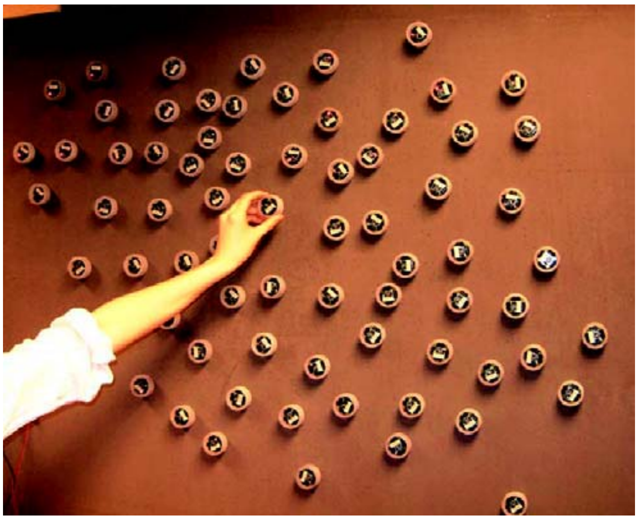


Fig 2 The Pushpin computer and its power plane.

larger sensor networks, in that it is a hardware test bed on which sensor network algorithms can be rapidly prototyped with realistic communication and sensing characteristics.

The Pushpin network consists of over a hundred nodes mounted on the powered substrate. A Pushpin node has a modular, stacked architecture comprised of four circuit boards, one each devoted to power, communication, processing, and sensing, as illustrated in Fig 4. As each layer is easily swapped out, the Pushpins are readily reconfigured (e.g. a collaborating group at the MIT Media Lab has developed an RF communications layer that swaps with the IR layer [47]). Each node receives power and ground through a pair of tensile pins located on the bottom of its power layer. These pins make contact with two parallel metal sheets embedded in a polyurethane foam sandwich [48], measuring 1.2 m by 1.2 m by 0.02 m. The Pushpin processing board contains an 8051-core, 8-bit, 22-MIPS microcontroller made by Silicon Labs (formerly Cygnal) [49]. This processor has 2.25 Kbytes of RAM and 32 Kbytes of non-volatile flash memory, as well as a host

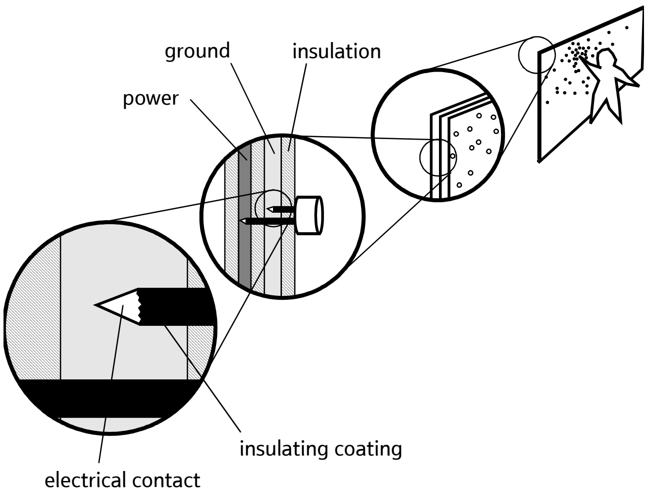


Fig 3 An array of Pushpins freely distributed atop their power-bearing substrate

Fig 4 Exploded view of Pushpin stack, showing all layers in configured order.

of on-board digital and analogue peripherals used for sensing and actuation.

The Pushpin processor transmits and receives data at 96 kbit/s using infra-red communication hardware on the communication module, where an IR transceiver points along each of the four cardinal directions, thereby enabling communication with neighbouring nodes. All four transmitting IR LEDs are hardwired in parallel and information as to which receiver is actively receiving a packet is available. In order to more evenly disperse the transmitted IR in all directions, a frosted polycarbonate ring is placed around each Pushpin. An earlier system developed at the MIT Media Lab in 1999 by Poor and collaborators [50] used a less compact conical mirror arrangement to omnidirectionally diffuse IR broadcasts across a similar network made from dozens of toy nodes called 'Nami'. Very little collaborative processing was done on this platform — its primary application used a capacitive sensor to detect the touch of a user's hand on a Nami node's translucent frame, which instigated a broadcast message that flooded the network and correspondingly changed the array's dominant colour via embedded LEDs.

repositioning a node is a trivial matter, hence the nodes can be readily reconfigured

Figure 5 shows the IR communication radius of different nodes in an actual Pushpin deployment. This figure shows that IR communications are local, which enables the Pushpins to become an exceptionally compact platform that realises a wireless sensor network. RF communication would be difficult to constrain at this short range; any node broadcasting RF would likely be received by the entire network.

The sensing module used here includes a phototransistor, 40 kHz ultrasound transducer, and electret microphone. The phototransistor and the ultrasound transducer were specifically included for use in localisation as described below, whereas the microphone is meant for general-purpose audio sensing in future applications. In addition to AC-coupled, amplified versions of all three raw sensor channels, enveloped versions of the ultrasound and audio microphones are available, as is the DC signal straight off the phototransistor. For actuation, the sensing module includes an RGB LED used to indicate the status of the node or as a display element when a Pushpin node participates as a 'smart pixel' in a distributed display task. Figure 6 shows a completely assembled Pushpin with all four modules but without the IR diffuser ring.

Programming an entire network of sensor nodes can be an onerous task. We have addressed this bottle-neck by creating a bootloader that always resides in the flash memory of every Pushpin. The bootloader performs version checking, error detection, and error correction on software updates received through the Pushpin IR communication hardware, and then writes the updates to bootable non-volatile program memory. We use a large 108-LED 'IR spotlight' that illuminates the entire power plane to beam updates to every Pushpin simultaneously from a desktop computer where new code is written and compiled. Using this interface, we can reprogram 100 nodes with entirely new operating system software in less than a minute, thus significantly reducing the time required for a debugging cycle. Once a software update is complete, the bootloader hands control over to 'Bertha', the Pushpin operating system. Bertha manages the hardware on the Pushpin, making basic services available to application code, which is included with Bertha during compilation, through a set of simple APIs. Some of the services provided by Bertha include a real-time clock, random number generation, sensor/actuator access, and access to interrupt routines.

One crucial service provided by Bertha is a comprehensive communication library for sending and receiving data packets. This library is built around two primitive packet types — an unacknowledged packet with no guarantee of delivery to neighbours, and an acknowledged packet with a high degree of certainty as to whether a transmitted packet was properly received. The difference between these packet types is similar to the differences between the UDP and TCP Internet protocols; unacknowledged packets are used for sending a large amount of data in cases where lost packets do not seriously hamper the performance of the system, while acknowledged packets are a more precise tool for sending critical data to a single network peer. In Bertha, reliable packets are implemented using an automatic repeat request (ARQ) protocol. Regardless of their type, all packet headers and data payloads are subjected to their own 8-bit cyclic redundancy check (CRC) to detect transmission errors.

In order to facilitate the addressing of packets to other nodes, each Pushpin has a network ID. These IDs need not be globally unique; in some sensor network applications, it is sufficient for a node to have an ID that is only unique among the set of nodes with which it can directly communicate. For example, a sensor node that is tracking a target only needs to know the ID of the immediate neighbour, which it will wake up when the

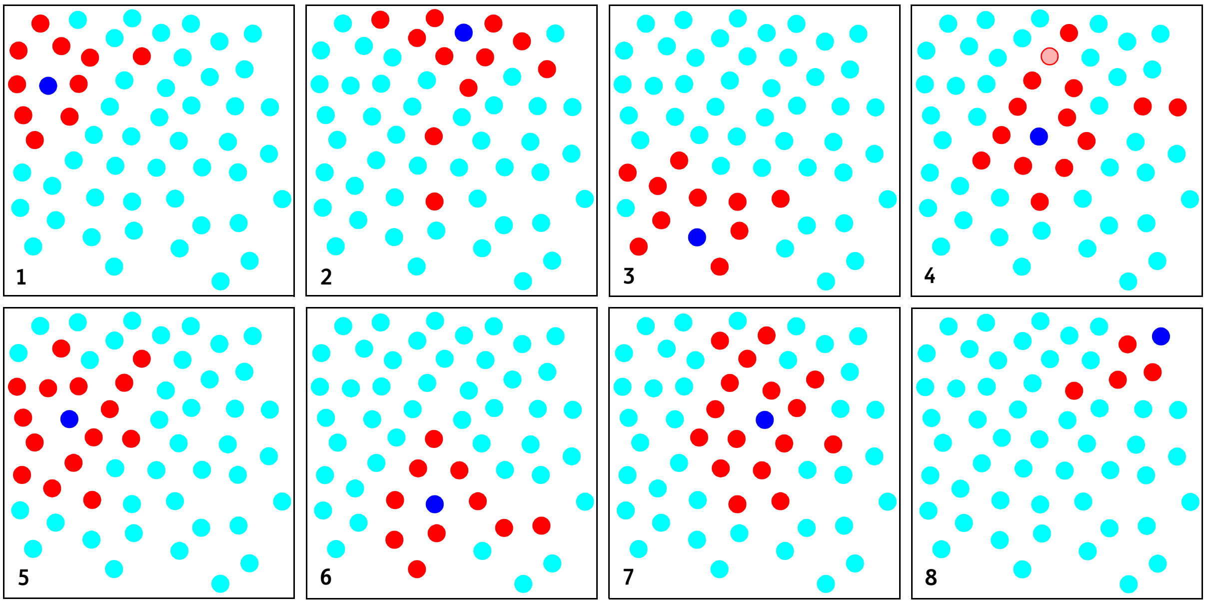


Fig 5 Network neighbourhoods for different Pushpin nodes. The transmitting node is coloured solid blue — all nodes that receive the messages are coloured red. All nodes received 100% of the messages, excepting the shaded node in example 4, which received sporadically. The irregular reception profiles are due to directional inhomogeneities in the IR profile, reflection and diffraction, and random line-of-sight optical paths between occluding Pushpins. The blue Pushpin was the only one transmitting in these tests — there was no other network traffic.

target enters that neighbour's sensing range. The Bertha communication library can automatically choose a random ID (based on sampled sensor noise) that is unique to the network neighbourhood of a node, or it can use a preprogrammed ID that has been individually assigned by the sensor network operator. In most of our studies, each node was programmed with a globally unique ID in order to simplify the task of gathering data from the sensor nodes to a desktop computer.

It is often necessary to send a broadcast to all nodes in the network. A broadcast contains a piece of data relevant to all nodes. This may be an element of information such as the co-ordinates of an anchor point (or reference location) in a

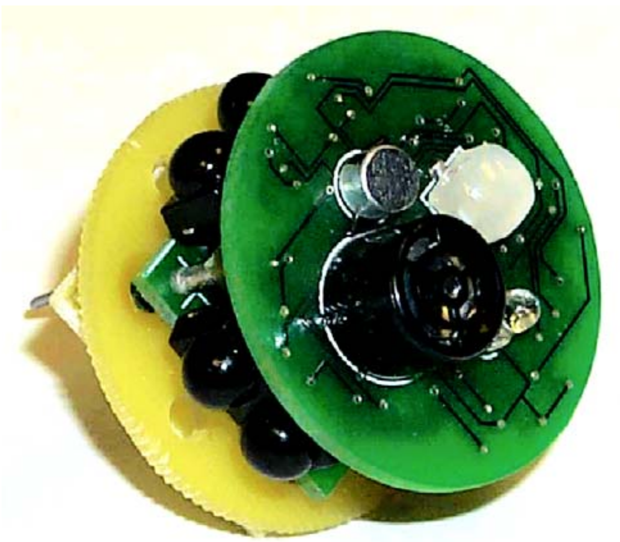


Fig 6 An assembled Pushpin stack, without IR diffuser.

distributed localisation algorithm, a command such as 'Stop what you are doing and prepare to be queried', or a request like, 'Will node X please broadcast its co-ordinates?' Because it is so widely used, this functionality is part of a special messaging layer of the communication library. A broadcast message is essentially directed diffusion [51]; a node chooses whether or not to retransmit a message based on the message hop count. If the hop count is less than or equal to the hop count from the last time the node transmitted this message, or if the message has not been seen before, the message is resent.

Similarly, the Pushpin system can be put immediately into a diagnostic mode upon the receipt of a unique sensor stimulus (e.g. two closely spaced light flashes). Data from individual nodes can then be routed through the network as described above, or the entire network can be queried in parallel via the IR spotlight used to program the nodes. We have developed a compact interface that can query any node that it approaches, allowing us to tap anywhere into the network. Additionally, an IR video camera looking at the Pushpin plane nicely images inter-node communication, providing valuable insight into network dynamics.

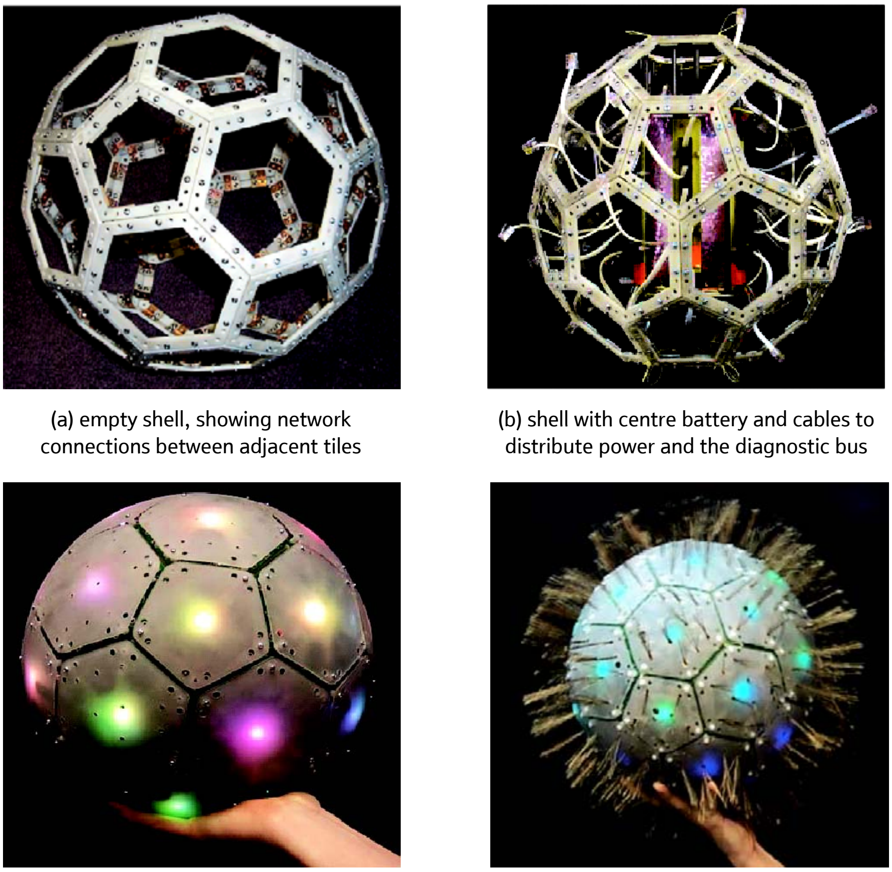
In order to utilise the Pushpin system for distributed sensor processing, the Pushpins need to know where they are located relative to one another. Although there are many techniques for localising sensor nodes [52], most are appropriate for much coarser node spacings, and do not provide the resolution needed for a network of this density. We had initially planned to utilise simple hop counts and establish gradient-based co-ordinates, as in Butera [40] and Nagpal et al [53], but, as indicated in Fig 5, the IR communication radius is much too wide and irregular to achieve an accuracy on the order of the node spacing. In order to achieve localisation at high density, we have built a hand-held device that produces a fast optical flash coincident with an ultrasound pulse and devised a set of distributed algorithms that process these signals in the Pushpin array. By measuring the delay between the flash (which we assume is instantaneous, and synchronises all nodes) and the ultrasound arrival (which propagates at the speed of sound), we have found that three bursts at different locations above the array are able to localise the sensor nodes in an absolute co-ordinate system to within a standard deviation of 2 cm [54]; ongoing updates to our sonar transmitter and signal processing systems promise to improve this result significantly.

Other studies in progress with the Pushpin system are exploring a variety of applications and issues of relevance to sensor networks. These include efficient algorithms for distributed image recognition (such as collecting a set of simple cumulants that can distinguish different shapes projected on the array), developing strategies for adapting to global sensor stimuli that adequately samples the activity without dominating inter-node communication, exploring implementations of robust node synchronisation schemes [55] and array colouring/segmentation algorithms [56] that promise to improve the reliability of communication between array elements and between the array and remote transceivers, and distributed imaging of acoustic transients [57—59].

3. Tribble — tiling an object with a sensor network skin

The other test bed that we have developed is called the ‘Tribble’, which here stands for ‘tactile reactive interface built by linked elements’. Introduced in Lifton et al [60], it is a coarse interpretation of electronic skins realised as a dense sensor network, as defined in section 1. Geometrically, Tribble resembles a large soccer ball, an inflated version of a polyhedron known as the truncated icosohedron, which is composed of 20 hexagonal faces and 12 pentagonal faces. Each of the 32 faces consists of circuit cards that can be considered as an individual ‘patch’ of skin. The patches screw into a plastic frame and can be individually removed and/or replaced at any time during Tribble operation. Figure 7 shows the Tribble in various stages of construction.

All of Tribble’s processing capabilities reside in distributed form via these patches; there is no central controller or master patch. Four NiCd D-cell batteries and accompanying voltage regulation circuitry are suspended at the middle of the frame, providing approximately 5000 mAh, distributed among all 32 patches via a star configuration of RJ22 cabling emanating outward from the centre. Tribble can also be powered by an



(a) empty shell, showing network connections between adjacent tiles

(b) shell with centre battery and cables to distribute power and the diagnostic bus

(c) working Tribble with all patches installed, but no whiskers

(d) completed Tribble with all patches and whiskers

Fig 7 Tribble, at various stages of construction.

external DC power supply for longer operation. Figure 7(b) shows these components.

The same RJ22 cabling also provides a global communications bus as a means of programming and debugging all patches in parallel from a personal computer; the global bus is not otherwise used by the patches themselves during normal Tribble operation. Rather, the patches communicate neighbour-to-neighbour — the same screws that mechanically secure a patch to the frame also provide a 115 200 bit/s communication channel to each of the five or six neighbouring patches via a direct electrical connection through conductive brackets fixed to the frame (see Fig 7(a)). These communication channels are fed through a multiplexer to the patch's 8-bit, 22 MIPS microprocessor (the same device used by the Pushpin system described in section 2), located on the underside of the patch. This microprocessor locally manages the patch's sensor data collection, actuator response and communication with neighbouring patches. For actuation, each patch has at its disposal a vibrating pager motor, a RGB LED, and a small speaker. As for sensing, each patch is equipped with 7 (pentagonal patches) or 12 (hexagonal patches) whisker sensors attached to a piezoelectric pick-up at their base, three FSR pressure sensors (enabling determination of the magnitude and centre of applied force), a solid-state temperature sensor, a light-dependent resistor (LDR) light sensor, and an electret microphone. The sensor suite is tiered appropriately to the length scales needed for physical embodiment — the microphones alert the Tribble to nearby activity, the light sensors see shadows from ambient illumination, hence detect proximity out to about 10 cm, the whiskers signal an approach within their 5 cm length (vital distance-judging cues to animals such as cats [61]), and the FSRs provide continuous contact pressure data.

Figure 8 details the patch elements. A hard, frosted polyethylene shell covers the exposed side of each patch; in addition to protecting the circuitry and diffusing the LED illumination, this shell distributes applied pressure across the three FSRs, to which it is mechanically coupled via foam elastomers. The whiskers consist of clumps of nylon/polyester paintbrush bristles protruding from small holes in the shell; beneath the shell, each clump is glued to a piezoelectric cantilever (the Minisense 100 from MSI[1]) soldered to the circuit board.

[1] See MSI Web site — http://www.msiusa.com/

All told, Tribble features 516 channels of 10-bit sensor input being sampled at approximately 1000 Hz per channel on average (actual sampling rates depend largely on the sensor being sampled). Thus, the Tribble has approximately 5 Mbit/s of aggregate sensory bandwidth (an empirically tested upper bound of Tribble sensory bandwidth is actually closer to 22 Mbit/s, but this is impractical as there would be no CPU resources left over to process the data). In comparison, the previously mentioned Robonaut humanoid telepresence robot has approximately 150 sensors on each of its two arms [19].

The software underlying Tribble is distributed among its 32 patches of 'skin', residing in the 32 Kbyte onboard flash memory of each patch's 8-bit microcontroller. All patches are loaded with essentially identical code, the only exceptions being a two-byte random seed and slight variations between hexagonal patch and pentagonal patch code due to the differing number of neighbours and whiskers. Although they all start out in similar states, the patches' behaviours quickly differentiate due to variations in their sensor data.

all of Tribble's processing capabilities reside in distributed form — there is no central controller

The software is divided into two main threads of operation — an interrupt-driven communication system and a sensing/actuation system. The communication system sequentially scans all the channels to neighbouring patches in such a way as to provide some guarantee of receiving packets regardless of which channel it is currently looking at when the neighbour starts sending. In any case, all communication between patches is predicated on a request/acknowledge protocol and an 8-bit cyclic redundancy check (CRC). In addition to the channel for each of its neighbours, a patch can monitor the global bus for commands issued to all patches in parallel from a desktop computer. Similarly, every patch is equipped with a stereo jack accessible through a hole in the outer shell, allowing each patch to be individually accessed from a desktop computer. Under normal operating conditions, neither of these channels is used; they exist primarily for debugging, uploading new code to the patches, and downloading data to a desktop environment.

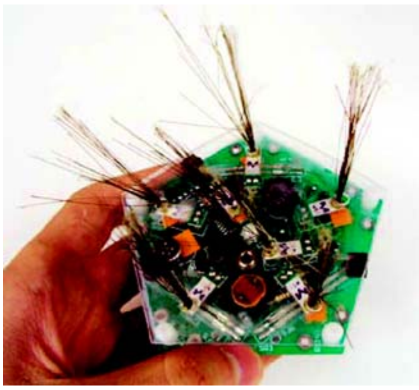


Fig 8 Skin 'patch' with transparent cover, showing sensor detail (right) and attachment of whiskers (left).

The bulk of the functionality and behaviour of Tribble emerges from the sensing and actuation system. The inherent trade-off to be made here is between the sampling rate of the sensors and the complexity of the output derived from the sensed data; a higher sampling rate necessarily reduces the number of CPU cycles devoted to processing the sampled data, determining what actions to take based on those results, and then executing the desired behaviour. This trade-off can be dynamically balanced in software by adjusting the aggregate sampling rate (samples per second per patch, summed over all channels), which can be set as low as 300 Hz and as high as 72 kHz. Once the aggregate rate is decided, it must be divided among all the sensing channels on the patch, as the different sensor modalities are suited to different acquisition intervals. This is also dynamically determined in software, such that all channels are sampled at a constant rate. Conflicting requests by different sensor channels to be sampled (there is only one ADC) are avoided by choosing relatively prime sample periods. As these parameters tend to change most slowly in HCI applications, our current implementation samples the temperature at 0.5 Hz, the light sensor at 4 Hz and the pressure sensors at 34 Hz. Much higher sampling attention (593 Hz) is devoted to the whiskers, in order to promptly catch transient events (it is well known that whiskers play a very important sensory role in mammals that possess them, where they tend to be both highly sensitive [62] and heavily linked to many areas of the brain [63]). As there is no dedicated hardware to determine the envelope or spectral components from audio, the microphone is sampled fastest, between 10—12 kHz. At these sampling rates, there are enough remaining processor cycles to manipulate the incoming sensor data in simple, yet useful ways. For example, windowed averages are kept for each sensor channel on each patch. Note that, although each sensing channel is sampled at 10 bits, for reasons of speed, only the most significant 8 bits are used.

Actuation of a patch's vibrating motor, RGB LED and speaker are mapped in software according to this information and executed in parallel with the sensing routines. The vibrating motor and each colour of the RGB LED are pulse-width modulated, allowing for a wide range of vibrotactile and colour feedback, respectively. The speaker can replay at 8 kHz any of approximately a dozen short 8-bit sampled sounds. Furthermore, additive sound synthesis and frequency modulation (FM) synthesis are used to create sounds with arbitrary timbre [64].

The Tribble is being used as a platform to simultaneously explore different research areas. At a low level, communication packets sent between patches simulate a model of chemical production, build-up, and decay as might occur in biological tissue. As in biology, the resulting chemical gradients form the basis of a distributed regulatory system, which in the Tribble's case, controls aspects of behaviour. Biological primitives such as lateral inhibition and excitation are emulated in this way as well [32]. We are using these analogies to develop algorithms that desensitise and inhibit sensor response when appropriate, in correspondence with adaptation behaviour in the human sense of touch. For example, the sensor should become accustomed to a light breeze and pay attention to more salient events. Similarly, we are exploring distributed entrainment, where the pseudo-chemical potentials accumulated in the patches lead the Tribble to anticipate particular stimuli and attempt to enhance or diminish its response. When interacting with a single user, the Tribble may build up large reactions that bleed across several tiles adjacent to the point of stimulus. When simultaneous stimulation is detected in several regions across the sphere (indicating many users), the Tribble's responses will become highly localised to maintain a degree of causality in its interaction.

The massively distributed sensing and actuation embedded within Tribble make for a unique platform to explore novel human-computer interfaces that exploit very high density, multimodal sensing and distributed, collocated actuation. For example, audio produced from 32 small speakers distributed atop a sphere [65] produces a very different experience than audio coming from standard computer speakers, especially when the user is close to the array. Tacto-audial interfaces for the visually impaired and controllers for musical installations represent potential applications.

In addition to acting as a test bed for distributed sensing and actuation, Tribble was designed to be an evocative object for interactive art venues, bringing the concept of skins as sensor networks to the general public. Figure 9 shows the Tribble on display at a robotic art show in New York City in July 2003 [66]. Here, the Tribble was suspended in the middle of an open rectangular frame, so it was easily accessible to direct manipulation by passers-by.

Fig 9 The Tribble on display to the public at the Artbots 'Robotic Talent Show' in July of 2003.

4. Z-Tiles — coating a floor with a sensor network skin

Another project within our sensate media initiative, the 'Z-Tiles', is a collaboration based in Ireland with the University of Limerick (Mikael Fernstrom, Bruce Richardson, and Krispin Leydon, and colleagues at the Interaction Design Centre) [67, 68]. This effort is aimed at covering a floorspace with a sensor network skin in order to dynamically image footsteps. Previous efforts that deployed sensor floors for interactive dance environments [69—71] or identification and context extraction for ubiquitous computing [72] used dense arrays of multiplexed sensors that resulted in a mess of wiring, which inhibited practical scaling beyond modest areas (e.g. 5 square metres). Other sensor floors have used common digital busses connecting large numbers of tiles in parallel [73], which can reduce the wiring problem, but also create critical single-point failure modes, where a single chattery tile can bring down the entire bus. Our solution (Fig 10) is to configure the tiles as a peer-to-peer mesh sensor network, where each tile confers with its neighbours to reduce the pixilated pressure data into a small set of finite parameters that are routed tile-to-tile to an external connection at the array's perimeter.

Figure 11 shows a working set of Z-Tiles. Each tile is covered with an array of 20 FSR pressure sensor 'prexels', and a network of five processors (again, similar Silicon Labs devices), which continually monitor all sensor values and handle network routing and processing between tiles. A thin, 2-mm layer of plastic protects the FSR array from damage. The tiles are shaped like puzzle pieces, hence they form a firm interlock when assembled. Sliding flea-clip connectors located at 4 positions around the tiles form interconnections between adjacent tiles when they are assembled, propagating the network and power. Each tile can accordingly communicate directly with its neighbours, and the fit is sufficiently tight as not to introduce excessive, non-sensed 'dead space' between tiles.

Upon detecting a change in prexel pressure, an ellipse is fitted to the pressure sensor distribution, crossing tile boundaries as required for footsteps that are not contained on a single tile. Resulting parameters (e.g. major/minor axes, pressure centroid, pressure magnitude, pressure asymmetry) are then routed tile-to-tile to an external connection [68]. The system has been designed to pass these parameters across sensate floors several metres on a side with a latency of 10 ms (including parameter calculation and routing), in order to enable real-time dance performance, which establishes the most stringent technical requirements. As the network is dynamic, failed tiles can be isolated (e.g. ignored) and routed around. Multiple connections to the outside world would also avoid routing log-jams, where data from a large sensate floor attempts to funnel through a single tile. This will be especially relevant for a large floor with many active occupants.

A sensate floor with a half-a-dozen interconnected tiles has been demonstrated to achieve the system's performance goals [68], while ongoing work [74] seeks to perfect routing performance and inter-tile footstep processing.

5. Conclusions and future directions

This paper has presented three ongoing projects that illustrate the concept of electronic skins implemented as dense sensor networks — a modular pressure-imaging floor with node spacing of 30 cm, a sphere tiled with an array of multimodal sensor nodes spaced at 15 cm intervals, and a sensate wall

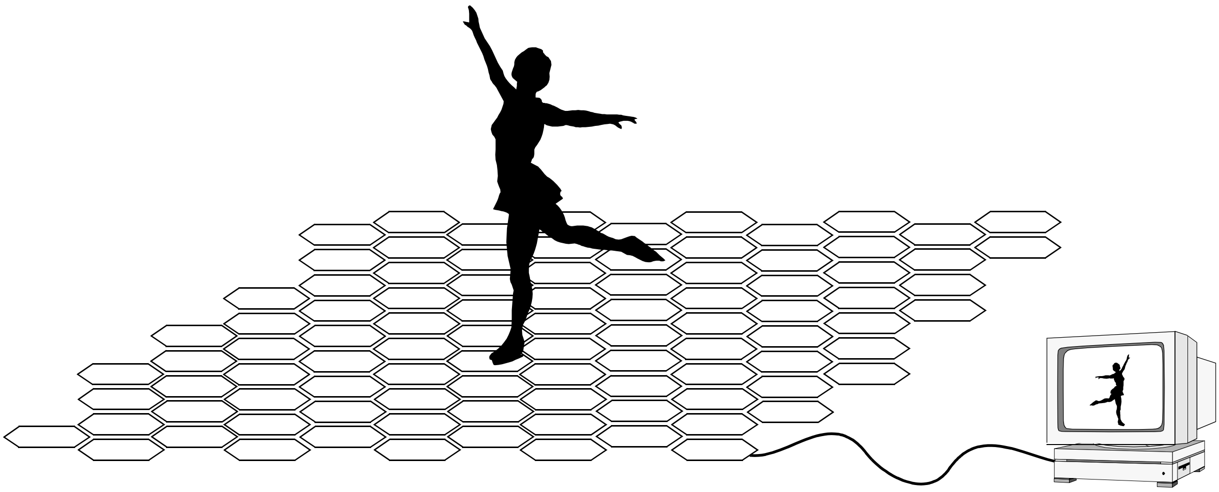


Fig 10 Schematic of the Z-tiles system. Interconnected sensate floor tiles form a pressure-measuring skin as a mesh sensor network.

Fig 11 Prototype Z-tiles hardware — (left) showing FSR sensors and processing electronics, and (centre/right) snapped together to form a sensor network.

panel with multimodal nodes that can be arbitrarily placed to within 5 cm of one another. We are currently using these test beds to develop and evaluate scalable algorithms for distributed sensor processing, and explore various applications for such densely sampled skins in fields such as human-computer interaction. Despite their technical complexity, the systems presented in this paper achieve a sensor density more than three orders of magnitude coarser than that of the human fingertip. Several steps can be taken to approach that level of detail.

An incremental step would be to use smaller devices (e.g. a processor similar to the devices used in this study is available in a 3 × 3 mm MLP-11 package) with denser packing of associated electronics all mounted on the rear of the skin nodes, together with much tighter integration of smaller sensor components on the top surface. While these techniques should enable a node spacing of better than 1 cm, avoiding the packages and moving to direct wire bonding of bare die, as used in multichip module (MCM) fabrication, could bring the density even higher, perhaps providing a node spacing better than 5 mm, depending on the sensor suite. The substrate could be a rigid multilayer printed circuit board for planar application, or a flexible circuit material such as thin Polyimide or Kapton foil, enabling conformal bending around a single axis.

These approaches, however, can become quite expensive with increasing device density, as each component needs to be explicitly bonded to the substrate. Current research, such as pursued by various groups for weather stations on a chip [75], seeks to integrate a diverse sensor suite (including whiskers [76, 77]) with a processor on a single die; such a device, with the appropriate sensors, could form a single, uniform component that could be bonded on to the skin substrate, alleviating much of the assembly complication.

Other fabrication techniques can be examined that promise to scale more economically. Some researchers are already working to build dense micromachined sensor arrays on large-area flexible substrates for electronic skins [17, 18]. Although these projects have succeeded in fabricating dense arrays of sensors to measure infra-red light, airflow, pressure, or shear force, none go very far to integrate electronics into the devices, especially digital processors, as the fabrication technologies are not entirely compatible.

Another process that holds promise for economic production of large-area electronic skins is printing — advances fostered by the display industry are evolving printed electronics [78, 79] and the hope of someday printing processors and radios [80]. Printed sensors are already a reality, e.g. pressure-measuring FSRs are made from piezoresistive and conductive ink, capacitive sensing electrodes can be made from printed carbon, and optical sensors can be made from printed material. Much of the technology behind printed electronics that is evolving quickly for flexible displays is very well suited to the fabrication of densely sampled electronic skins; indeed, some researchers are already moving in this direction [4, 8]. Another factor that is critical to successful scaling of sensate media is power dissipation. The Tribble system, which only contains 32 sensor nodes, quiescently consumes 2.5 watts. Although some of this drain is due to inefficient sensor conditioning electronics, the bulk of the power is taken by the processors. This indicates a problem when hundreds of processing elements are packed within a few centimetres of area — clearly the power per node needs to come down. There are several ways to approach this. Although an ongoing effort to perfect next-generation Pushpin Computers is implementing an even more powerful processor on each node [81], practical scaling to very high node density supports an opposite trend. The microcomputers used in our current systems are general-purpose devices used in an inefficient fashion. Adaptive, stimulus-dependent wake-up, clocking and sampling [82] and custom ASIC designs suited to the particular requirements of such densely sampled sensor networks [83], or hybrid analogue/digital architectures inspired by biological signal processing [84] are some approaches that can lead to much lower power requirements. Similarly, more sensors can be handled per processor, leading to fewer processors, for example, Lumelsky et al [5] anticipate an electronic skin with each processor handling an array of 10 000 sensors (assuming a StrongARM-class processor sampling all sensors at 60 Hz and devoting 1000 instructions per second to each sensor channel).

Finally, appropriate network topologies for electronic skins need to be evaluated. The prototypes presented in this paper are flat mesh structures, where all nodes talk only to their immediate neighbourhood. Biology, on the other hand, incorporates more of a tree structure, where signals are combined and processed as they evolve through neural pathways. As network loading can be quite dynamic in an electronic skin, an adaptive tree structure, such as being explored in macroscopic sensor nets [85], could be adopted, where a subset of nodes are able to talk to other nodes at larger distances, relieving local communication from the bulk of global data routing.

Acknowledgements

We wish to thank the Things That Think Consortium, and other sponsors of the MIT Media Lab for their generous support. Portions of this work are supported by National Science Foundation grant #ECS0225492. We also acknowledge the Z-Tiles collaboration with our colleagues at the Interactive Design Centre at the University of Limerick in Ireland (namely Mikael Fernstrom's team), which is supported by a grant from the Irish Government's HEA Foundation administered through the Media Lab Europe in Dublin.

References

1 Johansson R S and Vallbo A B: 'Tactile sensory coding in the glabrous skin of the human hand', Trends in NeuroSciences, 6, No 1, pp 27—32 (1983).

2 Aizenberg J, Tkachenko A, Weiner S, Addadi L and Hendler G: 'Calcitic microlenses as part of the photoreceptor system in brittlestars', Nature, 412, pp 819—822, (2001).

3 Bastian J: 'Electrosensory Organisms', Physics Today, 47, No 2, pp 30—37 (February 1994).

4 Lumelsky V J, Shur M S and Wagner S: 'Sensitive Skin', IEEE Sensors J, 1, No 1, pp 41—51 (June 2001).

5 Lumelsky V, Shur M S and Wagner S (Eds): 'Sensitive Skin' Singapore: World Scientific (2000).

6 Lee M H and Nicholls H R: 'Tactile sensing for mechatronics — a state of the art survey', Mechatronics, 9, No 1, pp 1—31 (February 1999).

7 Papakostas T, Lima J and Lowe M: 'Large area force sensor for smart skin applications', in Proceedings of the IEEE Sensors 2002 Conference, 2, pp 1614—1619 (2002).

8 Someya T and Sakurai T: 'Integration of organic field-effect transistors and rubbery pressure sensors for artificial skin applications', in the Proc of the 2003 IEEE International Electron Devices Meeting (IEDM), Washington DC, pp 8.4.1—8.4.4 (December 2003).

9 Xu Y, Tai Y, Huang A and Ho C: 'IC-integrated flexible shear-stress sensor skin', Technical Digest, Solid State Sensor and Actuator Workshop (SSAW), pp 247—250 (2002).

10 Richardson D: 'Smart skins and structures', Armada International, 22, No 2, pp 50—56 (April/May 1998).

11 Dietz P and Leigh D: 'DiamondTouch: a multi-user touch technology', in Proc of ACM UIST 2001, Orlando, Florida, USA, pp 219—226 (November 2001).

12 Rekimoto J: 'Smartskin: An infrastructure for freehand manipulation on interactive surfaces', in Proceedings of CHI 2002 Conference on Human Factors in Computing Systems, ACM, Minneapolis, pp 113—120 (2002).

13 Lee S K, Buxton W and Smith K C: 'A multi-touch three dimensional touch-sensitive tablet', Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI 85), San Francisco, CA, ACM Press, pp 21—25 (April 1985).

14 Westerman W and Elias J G: 'Method and apparatus for integrating manual input', US Patent No 6,323,846 (November 2001).

15 Omojola O, Post E R, Hancher M D, Maguire Y, Pappu R, Schoner B, Russo P R, Fletcher R and Gershenfeld N: 'An installation of interactive furniture', IBM Systems J, 39, No 3 and 4, pp 861—879 (2000).

16 De Rossi D, Carpi F, Lorussi F, Mazzoldi A, Scilingo E P and Tognetti A: 'Electroactive fabrics for distributed, conformable and interactive systems', in Proc of the 2002 IEEE Sensors Conference, IEEE Press, 2, pp 1608—1613, (2002).

17 Çelik-Butler Z, Butler D P, Yardanakul A and Yildiz A: 'Infrared sensors on flexible substrates for smart skin', SPIE Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AEROSENSE): Infrared Detectors and Focal Plane Arrays VII, Dereniak E L and Sampson R E (Eds), Orlando, FL, SPIE 4721, pp 260—268 (April 2002).

18 Engel J, Chen J and Liu C: 'Development of a multi-modal, flexible, tactile sensing skin using polymer micromachining', in the Proceedings of Transducers '03, 2, Boston MA, IEEE Press, pp 1027—1030 (June 2003).

19 NASA Robonaut project — http://robonaut.jsc.nasa.gov/

20 Stiehl W D: 'Tactile perception in robots: from the somatic alphabet to the realisation of a fully sensitive skin', BS Thesis, MIT Mechanical Engineering Dept (June 2003).

21 Hristu D, Ferrier N and Brockett R: 'The performance of a deformable membrane tactile sensor: basic results on geometrically defined tasks', IEEE Conference on Robotics and Automation, San Francisco, California, pp 508— 513 (April 2000).

22 Son J S, Monteverde E A and Howe R D: 'A tactile sensor for localising transient events in manipulation', in Proceedings of the IEEE International Conference on Robotics and Automation, pp 471—476 (May 1994).

23 Yamada Y, Maeno T, Fujimoto I, Morizono T and Umetani Y: 'Identification of incipient slip phenomena based on the circuit output signals of PVDF film strips embedded in artificial finger ridges', in Proceedings of the SICE Annual Conference, pp 3272—3277 (2002).

24 Kamiyama K, Vlack K, Mizota T, Kajimoto H, Kawakami N and Tachi S: 'GelForce: a vision-based force sensor for real-time measurement of force vector distributions', to appear in IEEE Computer Graphics and Applications (January 2005).

25 Hakozaki M, Hatori A and Shinoda H A: 'Sensitive skin using wireless tactile sensing elements', in Technical Digest of the 18th Sensor Symposium, Kawasaki, Japan, pp 147—150 (May 2001).

26 Yamada K, Goto K, Nakajima Y, Koshida N and Shinoda H: 'A sensor skin using wire-free tactile sensing elements based on optical connection', in Proceedings of the SICE Annual Conference, Osaka, Japan, pp 319—322 (August 2002).

27 Um D, Stankovic B, Giles K, Hammond T and Lumelsky V: 'A modularised sensitive skin for motion planning in uncertain environments', Proc of the IEEE International Conference on Robotics and Automation, 1, pp 7—12 (May 1998).

28 Wang P, Valencia P, Prokopenko M, Price D and Poulton G: 'Self-reconfigurable sensor networks in ageless aerospace vehicles', in Proceedings of the 11th International Conference on Advanced Robotics (ICAR-03), Portugal, Paper P100 (June—July 2003).

29 Cholewiak R W and Collins A: 'Sensory and physiological bases of touch', in Heller M A and Schiff W (Eds): 'The Psychology of Touch', Lawrence Erlbaum (Hillsdale, NJ), pp 23—60 (1991).

30 Pohja S: 'Survey of studies on tactile senses', Report R96:02, RWCP Neuro SICS Laboratory, Kista, Sweden (March 1996).

31 Howe R D: 'Tactile sensing and control of robotic manipulation', Advanced Robotics, 8, No 3, pp 245—261 (1994).

32 Brooks M: 'Highly redundant sensing in robotics — analogies from biology: distributed sensing and learning', in Tou J T and Balchen J G (Eds): 'Highly Redundant Sensing in Robotic Systems', NATO ASI Series, F58, Springer, Berlin and Heidelberg, pp 35—42 (1990).

33 Deutsch S and Deutsch A: 'Understanding the nervous system: an engineering perspective', IEEE Press, Piscataway, New Jersey (1993).

34 Gupta P and Kumar P R: 'The capacity of wireless networks', IEEE Transactions on Information Theory, IT-46, No 2, pp 388—404 (March 2000).

35 Scaglione A and Sergio S S D: 'On the interdependence of routing and data compression in multi-hop sensor networks', in the Proceedings of the Eighth Annual International Conference on Mobile Computing and Networks (MobiCOM 2002), Atlanta, GA, ACM Press, New York, pp 140—147 (September 2002).

36 Pradhan S S, Kusuma J and Ramchandran K: 'Distributed compression in a dense microsensor network', IEEE Signal Processing Magazine, 19, No 2, pp 51—60 (March 2002).

37 Gharavi H and Kumar S P (Eds): 'Special issue on sensor networks and applications', IEEE Proceedings, 91, No 8 (August 2003).

38 Hogg T and Huberman B: 'Controlling smart matter', Smart Materials and Structures, 7, pp R1—R14 (1998).

39 Abelson H, Knight T and Sussman J: 'Amorphous computing', MIT AI Laboratory Internal Report (1996) — http://www-swiss.ai.mit.edu/projects/amorphous/white-paper/amorph-new/amorph-new.htm

40 Butera W: 'Programming a paintable computer', MIT PhD Thesis, Media Laboratory, Massachusetts Institute of Technology, Cambridge MA (February 2002).

41 Sim H L: 'Scheme-based High-Level Simulator for Gunk (on the Wall)', — http://www.swiss.ai.mit.edu/projects/amorphous/hlsim/

42 NS2: 'C++ based discrete time network event simulator with TCP support', — http://www.isi.edu/nsnam/ns/

43 Swarm: 'Objective C based agent-based modeling toolkit', — http://wiki.swarm.org

44 Fang Q, Zhao F and Guibas L: 'Lightweight sensing and communication protocols for target enumeration and aggregation', in Proc of ACM Symp on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, pp 165—176 (June 2003).

45 Lifton J, Seetharam D, Broxton M and Paradiso J: 'Pushpin computing system overview: a platform for distributed, embedded, ubiquitous sensor networks', in Mattern F and Naghshineh M (Eds): 'Pervasive 2002', Proceedings of the First Pervasive Computing Conference, Zurich Switzerland, Springer Verlag, Berlin Heidelberg, pp 139—151 (August 2002).

46 Lifton J: 'Pushpin computing: a platform for distributed sensor networks', MIT MSc Thesis, Media Laboratory, Massachusetts Institute of Technology, Cambridge MA (September 2002).

47 Bletsas A and Lippman A: 'Natural spontaneous order in wireless sensor networks: time synchronisation based on entrainment', Technical Report, MIT Media Lab (Viral Networking Group) (December 2003) — http://web.media.mit.edu/~aggelos/papers/spontime.pdf

48 Dipline — http://www.saitec.fr/english/cadre.htm

49 Silicon Laboratories — http://www.silabs.com

50 Heaton K B, Poor R D and Wheeler A J: 'Nami', in SIGGRAPH 99 Conference Abstracts and Applications, ACM Press, New York, NY pp 215 (1999).

51 Intanagonwiwat C, Govindan R and Estrin D: 'Directed diffusion: a scalable and robust communication paradigm for sensor networks', in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000), Boston, Massachusetts, pp 56—67 (August 2002).

52 Bachrach J: 'Position determination in sensor networks', to appear in Stojmenovic I (Ed), 'Mobile *ad hoc* networking', John Wiley & Sons (2004).

53 Nagpal R, Shrobe H and Bachrach J: 'Organising a global coordinate system from local information on an *ad hoc* sensor network', in the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), Palo Alto, CA, Springer-Verlag, Berlin, LNCS 2634, pp 333—348 (April 2003).

54 Broxton M, Lifton J and Paradiso J A: 'Localising a sensor network via collaborative processing of global stimuli', to be published (2004).

55 Hu A and Servetto S D: 'Algorithmic aspects of the time synchronisation problem in large-scale sensor networks', to appear in the ACM Journal on Mobile Networks and Applications (2004).

56 Fitzpatrick S and Meertens L: 'Experiments on dense graphs with a stochastic, peer-to-peer colourer', in Gomes C and Walsh T (Eds): 'Probabilistic approaches in search', Workshop at Eighteenth National Conf on Artificial Intelligence (AAAI 2002), pp 24—28, AAAI Press (2002).

57 Allen D, Mitros P and Sussman G J: 'Acoustic aperture synthesis on an amorphous computer', MIT AI Laboratory Internal Report (Spring 2000).

58 Chen J C, Yip L, Hanbiao J E, Maniezzo W, Hudson D A, Yao R E K and Estrin D: 'Coherent acoustic array processing and localisation on wireless sensor networks', Proceedings of the IEEE, 91, No 8, pp 1154—1162 (August 2003).

59 Scott J and Dragovic B: 'Audio location: accurate low-cost location sensing', to appear in the UbiComp 2004 Adjunct Proceedings (2004).

60 Lifton J, Broxton M and Paradiso J A: 'Distributed sensor networks as sensate skin', in Proc of the 2003 IEEE International Conference on Sensors, Toronto, Ontario, pp 743—747 (October 2003).

61 Bush K: 'A true sense of touch', Catnip, 12, No 8, pp 3—5 (2004).

62 Dehnhardt G et al: 'Hydrodynamic trail-following in harbor seals (Phoca vitulina)', Science, 293, pp 102—104 (2001).

63 Moore C I: 'Frequency-Dependent Processing in the Vibrissa Sensory System', Journal of Neurophysiology, 91, No 6 (June 2004).

64 Roads C: 'The computer music tutorial', MIT Press, Boston, MA (1996).

65 Trueman D, Bahn C and Cook P: 'Alternative voices for electronic sound: spherical speakers and sensor-speaker arrays (SenSAs)', in Proc of the International Computer Music Conference, Berlin, San Francisco, International Computer Music Association, pp 248—251 (August 2000).

66 Artbots: Robot Talent Show (2003) — http://www.artbots.org/

67 McElligott L, Dillon M, Leydon K, Richardson B, Fernstrom M and Paradiso J A: 'ForSe FIElds — Force Sensors for Interactive Environments', in Borriello G and Holmquist L E (Eds): 'UbiComp 2002', LNCS 2498, Springer-Verlag Berlin Heidelberg, pp 168—175 (October 2002).

68 Richardson B, Leydon K, Fernstrom M and Paradiso J A: 'Z-Tiles: building blocks for modular, pressure-sensing floorspaces', in Proc of CHI 2004, Vienna Austria, ACM Press, New York, pp 1529—1532 (April 2004).

69 Paradiso J, Hsiao K, Strickon J, Lifton J and Adler A: 'Sensor systems for interactive surfaces', IBM Systems Journal, 39, Nos 3 and 4, pp 892—914 (October 2000).

70 Griffith N and Fernström M: 'LiteFoot: A floor space for recording dance and controlling media', in Proc of the 1998 Int Comp Music Conf (ICMC98), Ann Arbor, Michigan, International Computer Music Ass, San Francisco, CA, pp 475—481 (October 1998) — http://www.ul.ie/~pal/litefoot/index.htm

71 Pinkston R, Kerkhoff J and McQuilken M: 'A touch sensitive dance floor MIDI controller', Proceedings of the 1995 International Computer Music Conference (ICMC95), Banff, Canada, San Francisco, International Computer Music Association, pp 224—225 (September 1995).

72 Suutala J, Pirttikangas S, Riekki J and Roning J: 'Reject-optional LVQ-based two-level classifier to improve reliability in footstep identification', in Fresch A and Mattern F (Eds): 'Pervasive computing', Second International Conference, Pervasive 2004, Linz/Vienna, Austria, Springer-Verlag, Berlin, pp 182—187 (2004).

73 Eng K: 'Designing neuromorphic interactive spaces', PhD thesis, Swiss Federal Institute of Technology, (ETH) Zurich, Switzerland, Diss ETH No 15451 (2004).

74 Richardson B: 'Implementation of efficient data extraction from a scalable sensor system', MS Thesis, University of Limerick (August 2004).

75 Huang Q, Qin M, Zhang Z, Zhou M, Gu L, Zhu H, Hu D, Hu Z, Xu G and Liu Z: 'Weather Station on a Chip', Proc of IEEE Sensors 2003,Toronto, Canada, pp 1106—1113 (October 2003).

76 van Baar J J, Dijkstra M, Wiegerink R J, Lammerink T S J and Krijnen G M: 'Fabrication of arrays of artifical hairs for complex flow pattern recognition', Proc of IEEE Sensors 2003, Toronto, Canada, pp 332—336 (October 2003).

77 Chen J, Engel J and Liu C: 'Development of Polymer-Based Artificial Haircell Using Surface Micromachining and 3D Assembly', in the Proceedings of Transducers '03, 2, Boston MA, IEEE Press, pp 1035—1038 (June 2003).

78 Collins G P: 'Next stretch for plastic electronics', Scientific American, 291, No 2, pp 74—81 (August 2004).

79 Ridley B A, Nivi B, Jacobson J M: 'All-inorganic field effect transistors fabricated by printing', Science, 286 (5440), pp 746—749 (1999).

80 Mihm S: 'Print your next PC', Technology Review, 103, No 6, pp 66—70 (November—December 2000).

81 Butera W and Knaian A: 'V1.0 Dense Nixel Array', Personal communication (Summer 2004).

82 Benbasat A Y, Paradiso J A: 'Design of real-time adaptive power optimal sensor systems', to appear in the Proc of IEEE Sensors 2004, Vienna, Austria (October 2004).

83 Min R, Bhardwaj M, Cho S W, Ickes N, Shih E, Sinha A, Wang A and Chandrakasan A: 'Energy-centric enabling technologies for wireless sensor networks', IEEE Wireless Communications, 9, No 4, pp 28—39 (August 2002).

84 Sarpeshkar R: 'Analogue versus digital: extrapolating from electronics to neurobiology', Neural Computation, 10, No 7, pp 1601—38 (October 1998).

85 Kottapalli V A, Kiremidjian A S, Lynch J P, Carryer E, Kenny T W, Law K H and Lei Y: 'Two-tiered wireless sensor network architecture for structural health monitoring', in the SPIE 10th International Symposium on Smart Structures and Materials, San Diego, CA, Proc of the SPIE, 5057, pp 8—19 (March 2003).

Joseph Paradiso is Sony Career Development Professor of Media Arts and Sciences, directs the Responsive Environments group, and co-directs the Things That Think consortium. He has a diverse background, ranging from high-energy physics detectors and spacecraft control systems to electronic musical instruments. An expert on sensing technology, he has developed a wide variety of systems that track human activity using electric field sensing, microwaves, laser ranging, passive and active sonar, piezoelectrics, and resonant electromagnetic tags. A graduate of Tufts University, he received his PhD in physics from MIT as a Compton Fellow. Before joining the Media Lab, he worked at ETH in Zurich, and the Draper Laboratory in Cambridge, Massachusetts. He is the winner of the 2000 Discover Magazine Award for Technological Innovation for his expressive footwear system.

Josh Lifton is a PhD candidate in the Media Lab's Responsive Environments group.

His research focuses on how dense, distributed sensor networks can organise themselves on a local level to effect a global response to stimuli.

Before coming to the Media Lab, he attended Swarthmore College, where he majored in physics and mathematics, with a concentration in computer science and a minor in Chinese.

Michael Broxton is a master's student in electrical engineering and computer science at MIT.

He is currently a research assistant in the Responsive Environments group of the Media Lab, where he focuses on sensor network research. He helped develop the 'Pushpin' and 'Tribble' sensor network test beds, and is now exploring techniques for localising sensor nodes using global sensor stimulus.

He also recently spent a summer in the RADIANT networking group at Los Alamos National Laboratory where he designed a network analysis tool for tuning the performance of computational grids.