Large Surface Area Electronic Textiles for Ubiquitous Computing: A System Approach

David Graumann
Intel Corp.
david.graumann@intel.com

Meghan Quirk, Braden Sawyer, Justin Chong
Virginia Tech Dept. of ECE
{quirk, bsawyer, jchong}@vt.edu

Giuseppe Raffa
ARCES – University of Bologna
graffa@arces.unibo.it

Mark Jones, Tom Martin
Virginia Tech Dept. of ECE
{mtj, tlmartin}@vt.edu

***Abstract* – Electronic textile research often centers on the concept of introducing electronics to apparel such as shirts jackets, gloves, and health vests. Another less researched concept incorporates electronics into large textile surfaces such as carpets and upholstery. In this paper we explore methods and challenges of building a large surface area electronic textile floor for cooperative mobile device interaction. We systematically construct both a 100ft by 50ft textile simulation and a 3ft by 8ft working prototype using readily available materials. We introduce the broadly applicable embedded workload of human gait tracking as a means to defining the requirements for the textile physicals, networked computing node, and distributed execution environment subsystems. Through this effort we begin to establish a working model for combining inexpensive electronic textiles with a scalable execution environment that supports mobile device to floor interactions.**

I. INTRODUCTION

When we look at intersections of mobile computing and ubiquitous computing, we begin to envision environments where personal devices and anonymous environment-embedded services collaborate for an improved end-user experience. Our broader research is exploring how observations of human activities in a mobile setting combined with personalized information about end-user intent can be combined to better facilitate their desired task. The example we will use to illustrate this concept, and one which we believe generalizes to many other applications, considers pedestrian navigation within a building. An end-user carrying a personal device, such as an Ultra-Mobile PC (UMPC) or cellular phone, accesses a map of a building and establishes a few points of interest. Often what is provided today to guide the user to their desired destination is an on-device display and embedded location technologies. Though this is not a completely unreasonable combination of technologies to meet the user's goal, we assert that a far more efficient and natural approach is to establish a distributed navigation model, wherein the user offloads the guidance task to the floor by spawning a tracking agent that will guide them to their destination using a lit pattern on the floor. In this way, the user does not have to hold the device in view, nor relate the display to their real-world equivalent. Instead, their attention can be placed primarily on enjoying and discovering their environment and only occasionally and naturally glancing at the floor for direction and distance information. Their mobile device can now conserve its limited resources until richer content creation or control is needed. Location, orientation, and guidance are now being provided within the powered floor. The user can even establish simple human computer interaction with the guidance agent in the floor to pause, resume, and terminate the tracking agent by using foot patterns. By spawning an embedded floor application and establishing simple floor control, the battery-powered mobile device can now hibernate in their pocket until the richer interfaces are required. This is one of several user-centered concepts that involve mobile device and floor computing coordination.

As part of our research into unencumbered observations of human activity, and in light of the above usage scenario, we report here on our early findings from designing, building, and testing a textile-based carpet for sensing and communicating with occupants and their mobile devices. Our goal is to provide a large interactive surface area that has low cost, blends in seamlessly with the environment, scales from small rooms to large buildings, and is self-contained. While the device we describe is not itself mobile, it provides a fixed infrastructure for inexpensively deploying ubiquitous applications and supporting mobile computing devices. Existing flooring systems such as those described in [1][2][3][4][5] are expensive, do not scale well, only provide single sensor type input, and do not contain fully embedded computing. They either have low sensing resolution for large installations or high resolution for small area gait analysis in clinical settings. None of these systems are designed to support embedded and distributed computing applications to enhance mobile devices.

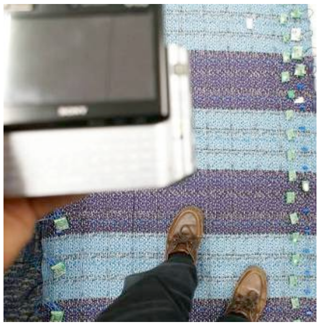

Figure 1. Ultra-Mobile PC associated through Bluetooth™ to electronic textile rug.

As an alternative approach, textiles are an attractive form for implementing this design for three main reasons. First, they cover large portions of our public and private living spaces in the form of carpets, drapes, and furniture coverings [6]. Secondly, existing manufacturing techniques in the textile industry are low-cost, and if it is possible to stay close to those techniques for pervasive computing infrastructures, then we will be able to exploit the economies of scale of the textile industry [7]. Finally, textiles provide a compelling fundamental structure for combining buses, sensors, power, and actuation over large surface areas. Today, incorporating electronics and computing into textiles results in a form of distributed embedded computing that occupies a very constrained corner of the design space, requiring examination of the methods of communication and computation to provide an execution platform that scales to a large surface [8][9]. What distinguishes this work from low-resolution large surface area plate sensing and high resolution small area gait sensing, is a system design that embeds high resolution sensing, wireless and lighted output, power, data, and computing functionality into textiles to instantiate a self-contained large surface area computing platform.

A. *Approach*

To better understand if floor systems with this level of tracking resolution and this breadth of mobile device collaboration can be realized, we investigated the use of electronic textiles as the cost-effective physical structure for realizing such a system. At this early stage a system level approach was selected to meet our initial objectives. We used gait tracking as the most broadly applicable embedded workload for applications using the floor observations. We selected a small Bluetooth™ module as the communication link between the device and floor. The textile designs, embedded distributed computing layout, sensing densities, and update rates were first simulated over a 100x50sq-ft area using a simulation environment first reported in [8]. This stage established system requirements. Then a 3x8sq-ft physical prototype rug was built to further explore the viability of such a system using low-cost materials. A variety of experiments on both the simulator and prototype rug were performed. Finally, a materials cost analysis was completed for first order approximation of the incremental cost associated with building this electronic textile floor covering.

B. *Design Criteria*

An important aspect of this research was to determine the feasibility and system requirements using very low cost textile materials and processing nodes over large surfaces. To set this approach apart from methods that others have pursued in the past, the following constraints were imposed

- The system must achieve low incremental cost over existing textile floor products.
- The materials used must be available in high volume.
- The system must have some form of localized end user output.
- The system must provide better location resolution than current mobile devices with a preferred resolution of 100Hz update rate and 2-inch square resolution.
- The input and output latencies must sustain the ability to track a 6ft/sec moving person.
- The execution environment must scale to arbitrarily large sizes.

C. *Paper Layout*

The remainder of this paper is organized as follows: Sections II-IV describe the system design. Sections V-VI present the performance results of the simulation and prototype. Sections VII-VIII summarize the paper and presents avenues for future work.

II. SYSTEM DESIGN

The main physical components of our system are: the raw textiles, the distributed processing nodes, and the communication channels. The main software components of our system are: the scalable execution environment and the embedded applications. Their designs are described in the order that they are integrated into a final solution.

A. *The Raw Textile Surface*

Woven materials are built by interlacing two yarns, warp and weft, at a 90 degree angle. The warp yarns run through the loom, similar to a printing press, while the weft yarns are laced in varying patterns within the warp to create a fabric. To build our substrate we used an 8/2 cotton rug warp, a worsted wool weft yarn, tinsel wire for power and communication, BEKAERT's BEKINOX® 12/2 stainless steel yarn for unencumbered foot sensing, and 0.9mm electroluminescent (EL) wire for outputs. The wire runs, consisting of four adjacent tinsel wires with a stainless steel wire on either side, are set into a 3-inch grid spanning the entire fabric. The EL wire was placed in sets of three at 6-inch intervals in the weft with a repeated pattern that exposed arrow-like triangles. This is the base pattern that repeats throughout the 3x8 sq-ft raw textile. Lastly, two additional tinsel wires were run along the outermost warp direction to deliver separate power to the EL wires.

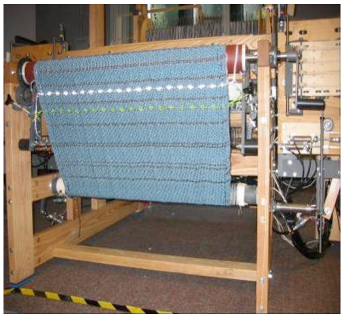

Figure 2. Initial rug on Virginia Tech's computer controlled loom.

Fig. 2 shows these materials in their woven state on the loom. The simplified weave pattern in Fig. 3 shows how weft and warp steel wires are separated by the wool fill material to create an inexpensive sensing surface. The surrounding four wool strands float the traveling steel wire above and below the surface just at their crossing. The weft steel wire was then driven with approximately 2VDC while the warp steel wire's voltage drop was measured under software control. By reading these crosshair values under compressed or released conditions, a crude impression of surface interaction can be obtained. These crosshairs are placed at a density of every 3 inches in both warp and weft directions. In the final design, a custom broken twill pattern was exclusively crafted to achieve the desired electrical characteristics while simultaneously stabilizing the textile weave.

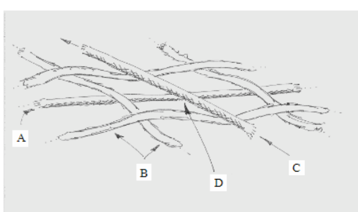

Figure 3. Resistive sensing in-fabric weave architecture. A & C steel wires. B wool fill & lift material. D sensing point.

Though more exotic materials and designs could have been selected, these yarns and structure created a useable and durable prototype using common manufacturing forms.

B. *Component Layout*

The loom outputs a structured arrangement of the steel, EL, and tinsel wires. At this stage, weft and warp directions are electrically independent except for the steel crosshair interaction. This final step establishes manageable regions within the textile. The layout selected, based on sensor densities, establishes a patch work of 3x3 sq-ft identical regions as shown in Fig. 4. This design can be realized with a single System on a Chip (SoC) residing within each region. The SoC must be physically connected to each of the steel wires in both the warp and weft direction, as well as connected to each of its adjacent

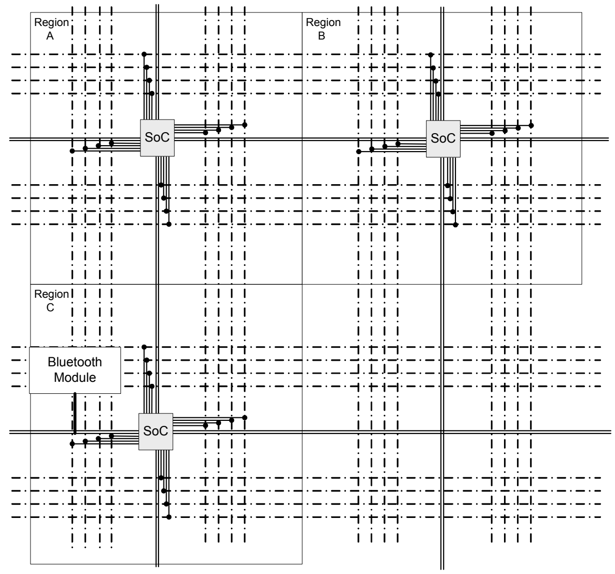

Figure 4. Textile electrical architecture. Regions A & B identical. Region C identical to A & B except additional Bluetooth™ module.

SoCs. We achieve these connections using an I2C bus running over a tinsel wire pair. At the desired density, a Bluetooth™ module is added to the I2C bus for routing information off the surface. One Bluetooth™ module may support more than one region. For the large scale 100x50 sq-ft simulation, we programmatically patch multiple regions together in both weft and warp directions. For the physical prototype we structured 2 2/3 3x3 sq-ft regions in the warp direction of the 3x8 sq-ft rug.

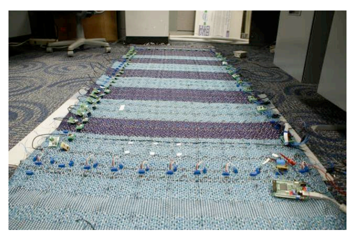

Figure 5. Final 3ft by 8ft textile prototype platform. Resistive sense(left edge). Resistive drive(top edge). Jumpered bus routers(bottom edge). CPU(bottom right). EL microcontrollers(right edge).

For the prototype, in place of the preferred SoC, we partitioned each 3x3 sq-ft region's system responsibilities across 30 ATMEL™ ATmega8L™ micro-controllers attached to the steel and EL wire and one Philips™ ARM7TDMI™ ARM™ CPU arbitrarily attached to the I2C bus within the region. Fig. 5 shows the final populated 3x8 sq-ft prototype. The bottom right corner holds the ARM™ and power supplies. Around the edges are the EL and steel wire microcontrollers which are managed by the ARM™. A single Bluetooth™ module is located halfway up the rug on the left side. Only one Bluetooth™ module is used at this point, though there is no reasonable limitation to their placement or density.

III. EXECUTION ENVIRONMENT

Unlike common sensing pads that route their data off the surface, this system is conceptually a large computing platform capable of self-contained interaction with the real-time dynamics of the occupants and their devices. To accomplish this, it is desirable for the embedded network of processing modules to establish a scalable execution environment that facilitates static and migrating embedded applications. The basis for this type of execution environment was the electronic textile *Softwear* publish/subscribe framework [10] originally designed and implemented for wearable textiles. In that work a set of processing nodes self-organize within a garment to support embedded applications in clothing. Our motivation is to determine if this framework will also support a large surface area textile containing both static and migrating applications. The execution environment is built upon 4 layers of functionality as shown in Fig. 6. Each is described below.

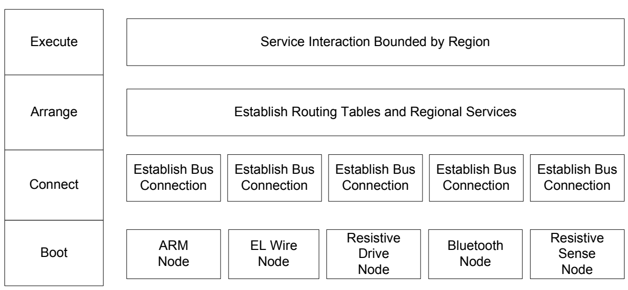

Figure 6. Execution environment layers

1) Boot: When powered on, each node initializes its hardware resources such as timers, analog-to-digital converters(ADC), and I2C serial hardware. Each node then configures a set of message queues to handle service messages between its hardware drivers and textile applications. These applications can reside within the node itself or on external nodes. In this way an Application Programming Interface (API) is provided for each of the hardware services within the textile regardless of where the application executes.

2) Connect: A router node is capable of communicating with at least two I2C buses. In a periodic scanning of each I2C bus, a query message is sent to all possible I2C addresses. Nodes respond to these messages with a series of messages specifying what services they provide. These responses contain the data to uniquely identify a node and its advertised services. A table of the nodes on each bus and their corresponding services is maintained within each region's router code. In this way, message flooding is avoided at power up because no broadcast messages are used to discover or advertise node services.

3) Arrange: Using the information obtained during the bus connection above, SoC nodes advertise their region's service information to their neighbors. These services are accompanied by a unique region identifier derived from the originating region's physical shape and location. By doing so, subscriptions can later be placed to location specific services even though these services are duplicated in each region. Layered on top of the routing table structure is the execution environment's publish/subscribe pairing mechanism which provides fulfillment methods for applications to access the textile services. All services (regardless of whether they are streamed crosshair values, wireless Bluetooth™ network modules, or output actuator lights) advertise their abilities to their region's centralize execution environment which conveys this to neighboring regions. In this way, a lightweight publish/subscribe structure is constituted that can be utilized by applications during the *Execute* stage below.

4) Execute: With the routing and advertised services established within each textile region, applications can now submit subscription requests to the system. This is done by sending a subscription request for a service within a specific region. Each SoC node contains the available services and routing information to direct this subscription request to the correct region. The application will receive back notification that the information is available from a publisher. Once this peer to peer pairing is made, an application utilizes a simple set of predefined messages provided by the *Softwear* framework. The messages used by the subscriber are 'onetime', 'repeat', and 'stop'. The 'onetime' message allows for simple command based control of the serivce. The 'repeat' method allows for a service to be repeated at a programmable interval. The 'stop' method allows for a 'repeat' operation to be halted and resumed without breaking the publich/subscribe pairing. With the slight application burden of requiring knowledge of the message structure and data format prior to exection, these three methods allow control of both actuation and sensing services in the textile.

IV. EMBEDDED APPLICATION

Within this execution framework, we implemented two primary capabilities and two secondary capabilities. The two primary capabilities are: steel crosshair input scanning and bi-ped occupant tracking. The crosshair scanning showcases a region's stationary application, while the tracking application showcases a migrating application. The two secondary capabilities are: EL wire output illumination and off-carpet wireless message passing via the Bluetooth™ module. The secondary features are output services in support of the two primary embedded capabilities and were implemented for system completeness. The scalability of these secondary features has yet to be explored in depth.

A. *Textile Surface Scan*

The textile surface was continuously scanned by sequentially driving each weft steel wire with 1.9VDC and then reading all of the warp steel wires of each 3x3 sq-ft region. A complete sequence through each weft set constitutes one scan. In this way the values from each crosshair are measured and provided to registered subscribers of that region. This application is duplicated in each 3x3 sq-ft region and does not migrate.

B. *Migrating Tracking Agents*

To reduce the bandwidth and latency build-up that occurs as the system scales to large surfaces, active crosshair clustering and bi-ped tracking algorithms were implemented to manage the simulation of two people walking over the system scaled to a 100ft by 50ft (100x50sq-ft) surface. Clusters were composed for samples approximating a 9-inch flat soled shoe, resulting in grouping up to 10 adjacent sensor readings into a foot descriptor. Cluster pairs within 4 feet were matched during initial classification and the pair was tracked with a Bayesian filter implemented using the Condensation filter described in [11]. This method was originally proposed to solve computer vision object tracking challenges and has been successfully applied in [5] for predicting the non-Gaussian uncertainties that occur in bi-ped foot placement. Under simulation, each of the surface's occupants received a dedicated filter consisting of 500 state particles representing the uncertainty of their weft and warp location, heading, and velocity. Euclidean distance was measured between the known location of the current processing unit and the tracking filter's output location estimation. When the distance grew larger than a threshold that indicated I2C bus bandwidth overload within the textile could occur, the tracking filter's internal state was migrated to the unoccupied processing node closest to the filter's reported occupant location. In this way, bandwidths and latencies between crosshair sensor readings, the tracking computing node, and EL wire illumination were kept low while the occupants traverse the large surface.

C. *Output*

The EL wires and the single Bluetooth™ module provided output to the user. Each EL microcontroller provided subscribing applications with direct control of the wire's illumination state allowing for the creation of loosely coupled sequencing patterns on the surface. The Bluetooth™ module provided a communication channel for the floor and the user's device to share data. In this case we used a simple ASCII terminal program on a UMPC to read from the Bluetooth™ serial port on the prototype rug as well as from a Bluetooth™ module attached to the computer running the large scale simulation. At this stage, we have yet to implement bi-directional communication through the Bluetooth™ module.

V. RESULTS

A. *Resistive Steel Mesh*

To assess the input sensing characteristics of the textile flooring, the overall resistive change under loaded and unloaded conditions was measured on a single crosshair and on combinations of crosshairs.

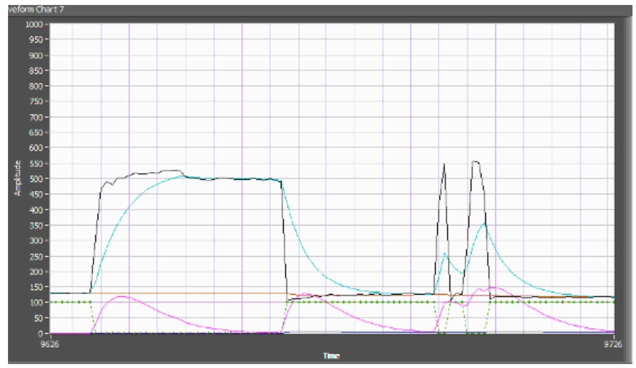

Figure 7. Single resistive crosshair test. Vertical axis: 10-bit ADC values. Horizontal axis: time. Darkest trace represents ball of foot press-hold-release(left) and press-release twice(right).

The single crosshair load condition was created by a male weighing approximately 160lbs pressing with the ball of one foot while wearing a flat soled shoe (See Fig. 7). The sensor isolation tests were performed by pressing directly on the crosshairs with a set of fingers (See Fig. 8).

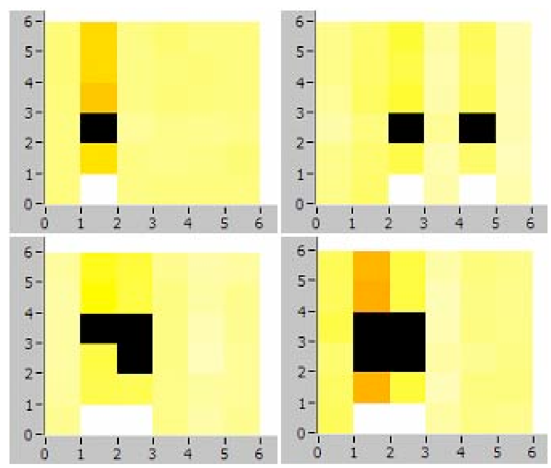

Figure 8. Crosshair isolation test. Intensity graph of resistive values over a 6x6 crosshair grid. Darker colors represent increased ADC values. Vertical is warp direction. Horizontal is weft direction. White through yellow represent crosshair noise floor. Labeled left to right, top to bottom: a,b,c,d.

1) *Steel Mesh Characteristics (physical prototype)*

- Typical unloaded-to-loaded Analog to Digital Converter (ADC) values covered 37% of full dynamic range (See Fig. 7).

- The unloaded crosshair noise floor drifted 10% of the full ADC dynamic range.
- Recovery time of an individual crosshair from loaded-to-unloaded was approximately 400 msec.
- Sensor isolation tests showed 1% dynamic range degradation of crosshairs in the warp direction. This degradation is considered insignificant (See Fig. 8 b,c).
- Sensor isolation tests showed up to a 50% loss in dynamic range for weft crosshairs (See Fig. 8 a,d). This degradation is considered significant. Although signal processing compensated for this phenomenon, we did not expect this behavior and additional analysis is needed.
- Typical foot pressure from a 160lb male wearing flat soled shoes registered ADC values under 10% of full dynamic range. This insensitivity significantly limited the prototype's ability to resolve gait characteristics.

B. Embedded Workload

Below are the performance characteristics of the workloads running within the surface. The stationary surface scan was measured using the physical prototype while the migrating tracking agent and overall system performance was analyzed under simulation.

1) Surface Scan Performance (physical prototype)

- A 69Hz scan rate was achieved for a single 3x3sq-ft region using a 100kbps I2C bus configuration and 8-bit ADC resolution. This rate is sufficient for bi-ped tracking, however, it is not sufficient for finer grain gait analysis applications and further improvements are needed to achieve the desired 100Hz rate.
- Scanned data consumed approximately 33kbps when two feet are placed on the surface, causing three occupants to exceed a single region's bandwidth.

2) Migrating Tracker Performance (simulation)

- A bi-ped tracking filter was able to track a pair of feet walking in a straight path across a 100x50 sq-ft simulation of the physical prototype using 500 particles that deployed 16-bit values for 2D location, velocity, and heading and 32-bit floating point values for their likelihoods.
- The tracking agent state migration consumed 9.06kbps using a state reduction from 500 to 100 particles as the initial input to the duplicated filter.
- A 3-bus maximum cost-function used for triggering state migration maintained a transfer rate of once per 1.3 seconds for a 6 ft/sec straight line walker. This is adequate but below the once per 1.5 to 3 seconds achievable.
- A once per second location output to the Bluetooth™ mobile device (at times spanning more than 3 buses) consumes < 200bps. This is for data describing location, speed, heading, measurement uncertainty, and a tracking agent ID. Additional information maintained by the tracker such as stride length, stance width, heel-to-heel timing, and shoe size was not included.
- The tracking filter's input sensor subscription method (for acquiring information from new regions) required additional bandwidth and complexity not in the original design. At this stage of development, the tracking application simply oversubscribes to all rectangular regions underneath its filter output location plus 9 feet. Significant refinement could reduce this area, which in turn would reduce the processing of erroneous data and bandwidth consumption under multiple occupancy conditions.

3) Scaled System Performance (simulation)

- Table I shows simulated bandwidth limitations for surfaces ranging from a hallway carpet to a larger office space. It shows that for smaller surfaces, the complexity of the system being designed is not required and raw data can be moved off the surface at an arbitrary point for further processing by a UMPC or centralized backend server. However, as the surface area increases, the ability to move data off the surface becomes bandwidth limited. Embedding active sensor clustering and migrating tracking techniques into the surface allowed the inexpensive buses to support the data flow while still maintaining surface scanning message movement (See Table I).

TABLE I. BUS BANDWIDTH OVERLOAD.

Surface Size	Raw	Cluster	Track
hall carpet (3x10)	Pass	Pass	Pass
bedroom (10x12)	Overload	Pass	Pass
family (15x18)	Overload	Overload	Pass
lobby (25x25)	Overload	Overload	Pass
office (100x50)	Overload	Overload	Pass

- Under simulation, we modeled the added latency of a location message moving through the routers attached to a set of four 100kbps I2C buses. Data movement was modeled for $1/9^{th}$ of the crosshairs of any 3x3 sq-ft region simultaneously collecting data. These latencies were shown to build to

600msec before leaving the surface due to I2C bus contention and built up message queues in the system. These messages are used by the Bluetooth ™ module for transmitting to the UMPC. Similar latencies were experienced when sending EL wire control messages through the system.

- Fig. 9 below represents a bird's eye view of the tracking algorithm running in the 100x50 sq-ft simulation. The two filled dots represent the tracked location of the two occupants. The gray cluster of smaller dots surrounding each location represents the weights and location of the migrating filter's state variables. A straight line tail extends from each filled dot to the center of a square, which represents the physical SoC being used to execute the filter's code. As each simulated occupant traverses the surface, the sensing inputs physically separate from this computing node and the tail elongates. Once this distance exceeds three bus lengths, the filter state is transferred to the idle SoC nearest the filter's reported location. The simple APIs provided by the execution environment completely supported this complex operation.

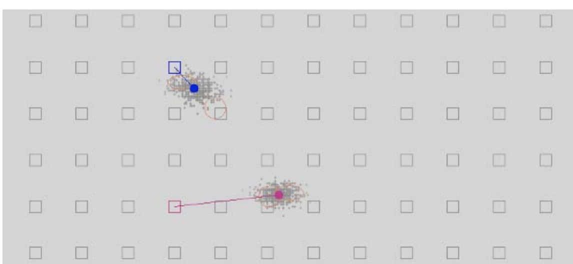

Figure 9. Migrating tracking filters. 18x36sq-ft bird's-eye view of two walking occupants - cropped from the 100x50 sq-ft simulated surface.

C. System Limitations

Though we achieved our initial goal of implementing and understanding the main functionality of this novel computing platform, a few of our hardware subsystems have significant limitations as identified below.

- The I2C controller is limited in its ability to monitor all network traffic. We easily added the ability for a single chip to read all bus addresses in the simulation code, but were forced to consume additional bus bandwidth to add routing path addresses in the I2C data packets.
- The Bluetooth™ discovery and association model is cumbersome and slow when attempting the fast seamless surface roaming in the target use cases. We used this radio because it is readily available and allows us to demonstrate the movement of a location data type from within the embedded system to the UMPC concurrent with the embedded scan running on the same bus. An alternative radio needs investigation.
- The physical prototype's lack of crosshair sensitivity, under occupant load, limited the ability to track gait level characteristics at a 3-inch granularity as originally intended. Additional signal processing and a modified weave pattern are under investigation.

VI. Incremental Cost

A first order approximation of the cost of building this system using a high volume Bill of Materials (BOM) was calculated from supplier quotes [12]. The wool fill material was omitted because we are only interested in the incremental cost of adding the buses, sensing wires, actuation wires, and computing nodes.

Before running the cost analysis, the current design was altered as follows to better depict a final solution. First we assume that the data bus and the power are distributed over the same tinsel wire pair. Second, we increase the steel wire density from a 3 inch to a 2 inch separation to bring us to the desired gait recognition resolution. Lastly, the few discrete resistive and capacitive components were omitted (See Table II).

TABLE II. Bill of Materials

Component	Comment	Unit $ Cost	Unit/ SqFt
stainless steel	BEKINOX®	0.04 /ft	12.00
tinsel	2 wire combined power & data	0.25 /ft	1.30
electroluminescent	Optional	0.50 /ft	0.70
micro-controllers	ATMEGA8™	1.97 ea	0.30
low power SoC	approx. based on XScale™	4.00 ea	0.03
connectors		0.10 ea	0.30
Bluetooth module		3.00 ea	0.01
Total BOM			1.93

Using these numbers the approximate added component cost is slightly less than $2.00USD per square foot. Additional cost would be incurred when attaching the computing nodes to the textile and testing for functionality. Then, relying on self-organizing, self-healing electronic textile designs reported in [7][8], it seems conceivable that final installation costs could be held comparable to that of conventional carpets.

VII. Future Work

Having established some level of functionality across all subsystems in this initial work, the primary research focus now becomes the optimization of hardware and software methods in each subsystem. The top two hardware and top two software maturation steps are:

A. *Physical prototype*

- Establish short-range radios capable of fast hand-off of the traversing mobile device.
- Explore alternative weave patterns for increased crosshair sensitivity under occupant load.

B. *Execution environment:*

- Exploit the redundant bus pathways through the textile surface for improved data movement and fault tolerance.
- Improve tracking methods for occupant isolation under a wide range of movement.

VIII. Conclusion

We demonstrated the ability to harness the inherent structure of a woven textile for building a ubiquitous computing platform capable of interacting with both the user and their mobile device. Readily available electronic yarns, small custom processing modules, and simple communication links showcased this technology. The surface crosshairs were electrically characterized and the performance of one stationary embedded application was measured. Simple EL wire lit an output pattern based on surface interaction. To complete the collaboration between the floor and a mobile device, a Bluetooth™ radio module was used to route floor information to the occupant's device. Scan rates and crosshair sensitivity of this initial physical prototype, though promising, fell just shy of our goal to resolve the higher resolution gait characteristics. The system design was scaled to larger dimensions in simulation to further demonstrate the viability of the physical structure, execution environment, and migrating algorithms to support the usage descriptions. A cursory assessment of the material cost indicated that this approach would not add an exorbitant price increase due to component cost. This met initial design goals, though the manufacturing costs for fully automated high volume processes remain unknown. To evolve our early realization of an inexpensive large surface area electronic textile computing platform, we will further explore weave patterns, novel radios, and routing hardware for seamless interaction with mobile devices.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CCR-0219809, CNS-0447741, and CNS-0454195.

References

[1] R. Glaser, C. Lauterbach, D. Savio, M. Schnell, S. Karadal, W. Weber, S. Kornely, and A. Stohr, "Smart Carpet: A textile-based large-area sensor network," http://www.future-shape.de/publications_lauterbach/SmartFloor2005.pdf, Accessed May 2007.

[2] Tekscan™, website, http://www.tekscan.com, Accessed May 2007.

[3] Gaitrite™, website, http://www.gaitrite.com, Accessed May 2007.

[4] L. Middleton, A.A. Buss, A. Bazin, and M.S. Nixon, "Floor sensor system for gait recognition," in Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies, Oct. 2005, pp. 171-176.

[5] T. Miyashita, M. Shiomi, and H. Ishiguro, "Multisensor-based human tracking behaviors with Markov Chain Monte Carlo methods," in Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots, Volume: 2, November 2004, pp.794- 810.

[6] D. Marculescu, R. Marculescu, N. Zamora, P. Stanley-Marbell, P. Khosla, S. Park, S. Jayaraman, S. Jung, C. Lauterbach, W. Weber, T. Kirstein, D. Cottet, J. Grzyb, G. Tröster, M. Jones, T. Martin, and Z. Nakad, "Electronic Textiles: A platform for pervasive computing," in Proceedings of the IEEE, vol. 91, no. 12, December 2003, pp. 1995-2018.

[7] T. Martin, M. Jones, J. Edmison, and J. Shenoy, "Towards a design framework for wearable electronic textiles," in Proceedings of the Seventh International Symposium on Wearable Computers, White Plains, NY, October 21-24, 2003, pp. 190-199.

[8] T. Martin, M. Jones, J. Edmison, T. Sheikh, and Z. Nakad, "Modeling and simulating electronic textile applications," in Proceedings of the 2004 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems, Washington, D.C., June 11-13, 2004, pp. 10-19.

[9] Z. Nakad, M. Jones, and T. Martin, "Communications in Electronic Textile Systems," in Proceedings of the 2003 International Conference on Communications in Computing (CIC 2003), June 2003, pp. 37-43.

[10] C. Zeh, "Softwear: A flexible design framework for electronic textile systems," Master of Science Thesis, Virginia Polytechnic and State University, April 10, 2006

[11] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density", in Proceedings of the European Conference on Computer Vision, Cambridge, UK., 1996, pp. 343-356.

[12] Digi-Key™, pricing quote communication, Website, http://www.digikey.com, Accessed May 2007.