Gestures Everywhere: A Multimodal Sensor Fusion and Analysis Framework for Pervasive Displays

Nicholas Gillian[1], Sara Pfenninger[2], Spencer Russell[1], and Joseph A. Paradiso[1]
{ngillian, saras, sfr, joep}@media.mit.edu

[1]Responsive Environments Group
Massachusetts Institute of Technology Media Lab, Cambridge, MA, USA
[2]Wearable Computing Laboratory, ETH Zurich, Switzerland

ABSTRACT

Gestures Everywhere is a dynamic framework for multimodal sensor fusion, pervasive analytics and gesture recognition. Our framework aggregates the real-time data from approximately 100 sensors that include RFID readers, depth cameras and RGB cameras distributed across 30 interactive displays that are located in key public areas of the MIT Media Lab. Gestures Everywhere fuses the multimodal sensor data using radial basis function particle filters and performs real-time analysis on the aggregated data. This includes key spatio-temporal properties such as presence, location and identity; in addition to higher-level analysis including social clustering and gesture recognition. We describe the algorithms and architecture of our system and discuss the lessons learned from the systems deployment.

1. INTRODUCTION

Modern digital displays are now equipped with a myriad of sensors to detect a user's actions. These sensors include cameras [16], capacitive touchscreens [3], infrared proximity sensors and beyond [13]. As these displays become pervasive throughout our workplaces, retail venues and other public spaces, they will become an invaluable part of the rapidly growing sensor networks that observe every moment of our lives. While these sensor networks are becoming ubiquitous, they are often deployed as independent closed-loop systems, designed using a single technology for a single application [12]. Consequentially, multiple systems frequently observe the same events, but fail to accurately detect these events because of occlusions, sensor noise or gaps in sensor coverage. To truly take advantage of these pervasive networks, we need dynamic frameworks that support the heterogeneous sensor data from nearby devices to be shared and fused. This aggregated data can then be used to make better inferences of higher-level information, such as a user's precise location or the classification of a gesture, which can then be distributed back to nearby pervasive applications through reusable software abstractions. By developing frameworks for sensor fusion and the abstract dissemination of analytical data, we can maximize the heterogeneous sensor networks distributed throughout our environments, and provide better inference on the events within range of these sensors; ultimately facilitating improved user experiences on both interactive displays and devices within their vicinity.

In this paper, we describe *Gestures Everywhere*, a real-time system for multimodal sensor fusion, pervasive analytics and gesture recognition. Gestures Everywhere (GE) is distributed across 30 digital displays located in key public areas of the MIT Media Lab and aggregates the real-time data from approximately 100 sensors that include RFID readers, depth cameras and RGB cameras. Our system fuses this multimodal sensor data from disparate sensors across different locations using radial basis function (RBF) particle filters and performs real-time analysis on the aggregated data. This analysis includes key spatio-temporal properties, such as presence, proximity, location and identity; in addition to higher-level analytics, such as social clustering and gesture recognition. This data is then made accessible to provide pervasive analytics and gesture recognition support to the interactive displays and other ubiquitous devices that run throughout the building. The main contributions of our work include: (i) a flexible framework for multimodal sensor fusion using RBF particle filters; (ii) a modular infrastructure for the abstract dissemination of fused analytical data; (iii) the integration of these frameworks into a robust, scaleable, customizable system that supports interactive displays and pervasive devices within their vicinity. First, we provide a general overview of our system followed by a review of prior sensor fusion work and a description of the RBF particle filter. In Section 4, we outline how the RBF particle filter is applied to estimate spatio-temporal metrics, such as presence and location. Section 5 presents how this fused data is then applied to make better inferences of higher-level information, such as gesture recognition. Finally, Section 6 outlines what we have learned from the development of this system and Section 7 concludes.

2. SYSTEM ARCHITECTURE

Gestures Everywhere is built on top of an existing interactive-information system [3], which has been running throughout the Media Lab since early 2010. This system, referred to as the "Glass Infrastructure", consists of over 30 digital-information displays distributed at key locations throughout the building. Each 46-inch Samsung display features an integrated capacitive touch screen and has been extended with

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
Perviasive Displays 2014 Copenhagen, Denmark
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2952-1/14/06 ...$15.00. http://dx.doi.org/10.1145/2611009.2611032

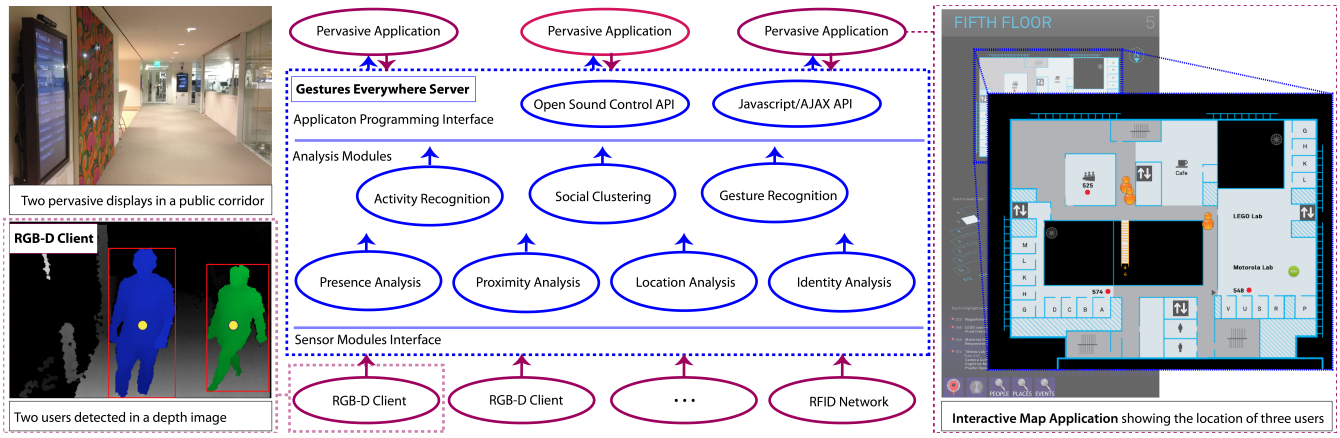

Figure 1: From left to right: (i) two pervasive displays located in a public corridor at the MIT Media Lab; (ii) two users detected in front of a pervasive display by the RGB-D client application, showing each user's center of mass and bounding box; (iii) an overview of the Gestures Everywhere architecture; (iv) the interactive map application visualizing the real-time location of three anonymized users.

a ThingMagic ultrahigh-frequency (UHF) RFID reader and custom sensor node [13] containing microphones, motion, temperature and humidity sensors. An Apple Mac mini, embedded behind each display, is used to receive events from the capacitive touch screen and to run an RFID-reader polling application written in Python. The Apple Mac mini also runs each display's user interface, built with HTML, CSS and JavaScript, running in fullscreen-portrait mode inside a WebKit-based browser. The displays provide continuous place-based information about the lab's research in addition to interactive building floor plans, event posters and event-centric applications.

Building on the Glass Infrastructure, we added 25 Microsoft Kinects to key displays located throughout the Lab. A custom-built C++ software application, running on the Apple Mac mini driving the associated display, manages each Kinect. This software application (RGB-D client) captures the depth and RGB images from the Kinect at 30 frames per second, processing each image locally using a suite of computer vision and machine-learning algorithms. These algorithms extract features for presence estimation and people detection from the RGB images, in addition to segmenting a user and estimating their overlapping skeleton model from the depth data[1]. The RGB-D client streams these features to a central server via the Open Sound Control (OSC) network protocol [21]. This includes basic statistics such as the number of users currently visible at a display, to more complex features such as the three-dimensional position of each detected user's centre of mass (see Figure 1), the color histogram values of a specific user, or the estimated location of the 24 joints of a user's body.

At the core of the GE architecture, a central server receives the real-time depth and image feature data from the RGB-D clients. This is in addition to the real-time data from the network of RFID readers located throughout the building. To maximize this heterogeneous data, the GE server fuses the data using a a series of hierarchical analysis modules. First, low-level analysis modules use RBF particle filters to fuse the multimodal data from nearby sensors. Each low-level analysis module computes a specific spatio-temporal property, such as presence, proximity, location or identity. The output of these lower-level modules are then fed into higher-level modules, which perform more complex analysis such as social clustering or gesture recognition. Together, these modules determine whether, which, and how users are interacting with a particular display and are used by a host of applications running on the GE infrastructure (see Section 6). To make both the fused data and analysis results accessible to other displays and devices, the GE system provides application programming interfaces (API), implemented in both OSC and HTTP/JSON[2]. These APIs act as an abstraction layer that supports pervasive applications to query, for instance, the current location of a specific user or the number of individuals within the proximity of a display. Once queried, the system will return the best estimation possible (using the fused sensor data from disparate sensors at that location); paired with a probabilistic representation of the system's confidence for that estimation. This enables client applications to respond accordingly, based on the system's estimation confidence.

3. SENSOR FUSION

In this section, we describe the RBF particle filter algorithm used for sensor fusion throughout the GE system. First, we describe other related work in sensor fusion.

3.1 Related Work

Sensor fusion has received extensive research across a diverse range of fields in computer science and engineering, including aerospace [1]; robotics [18]; people tracking [17, 5]; and pervasive computing [6]. Bayesian-based techniques, such as Kalman filters [4], and particle filters [6, 18], are particularly useful as they provide a powerful statistical tool to help manage measurement uncertainty and perform multi-sensor fusion. Particle filters' main advantages over other

[1]We use the OpenNI and NITE libraries for skeleton fitting, however all other metrics are estimated directly from the depth images so we can detect and track a user from the instant they enter the depth field.

[2]You can view the real-time output of the GE system via the Gestures Everywhere website: `http://ge.media.mit.edu`

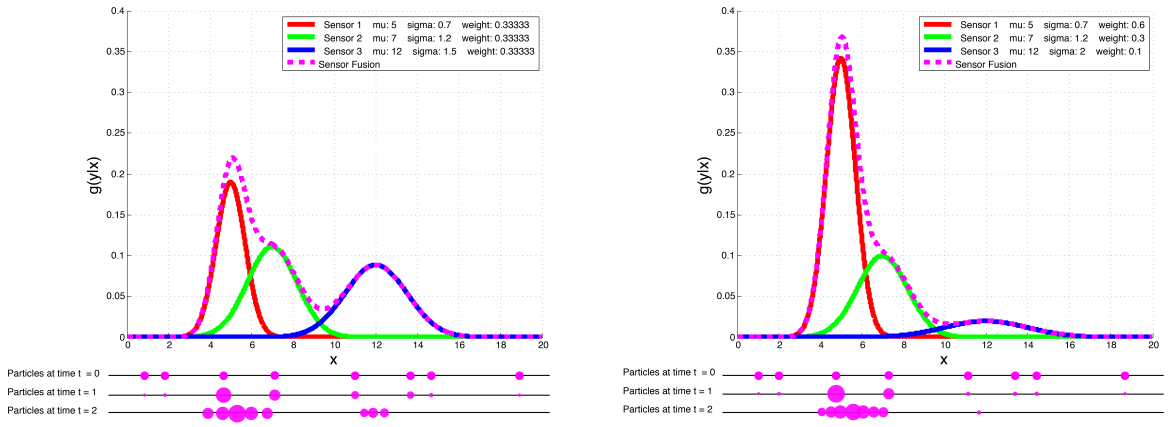

Figure 2: An illustration of fusing the output of three sensors using a Gaussian RBF. The left-hand graph uses equal weights for each sensor, however the right-hand graph places more weight on the first and second sensors. Note how changing these weights impacts the resulting likelihood function. The distribution of particles are shown beneath each graph for $t = 0, 1, 2$. Particles at time $t = 0$ are randomly distributed. The size of each particle reflects that particles' weight, with larger particles having a larger weight.

Bayesian techniques are their ability to represent arbitrary probability densities and that they can incorporate virtually any sensor type [6]. A common approach to sensor fusion using a particle filter is to model the sensor fusion process as the product of independent conditional probabilities across multiple sensors [11]. This approach has been applied widely to track and identify users by fusing inertial measurement units and WiFi [20]; vision and RFID data [7]; or a network of fixed cameras [2]. While this approach provides a convenient method to combine data from multiple sensors, it does have one disadvantage in that any sensor with a conditional probability approaching zero, will drive the product of the remaining sensors towards zero. Consequently, one damaged or noisy sensor could potentially suppress other functioning sensors which can make it difficult to implement a robust model that can be abstracted across a large sensor network. To mitigate this problem, RBFs can be combined with particle filters, as this provides an efficient algorithm to combine the sensor data from multiple heterogeneous sources, while remaining robust to unresponsive sensors. Radial basis functions have been applied successfully in several contexts with particle filters, such as RBF Neural Networks for sensor fusion [5] and target tracking [19], or using RBFs as an additional stage for interpolating sparse particles [14]. To provide a flexible framework for sensor fusion, we integrate the RBF directly into the computation of the sensor's likelihood model as we now describe.

3.2 Radial Basis Function Particle Filter

Particle filters are a set of on-line posterior density estimation algorithms that probabilistically estimate a dynamic system's state from noisy observations. The system's state vector could be the one-dimensional presence estimate of a digital display, or a more complex vector including the position, pitch, roll, yaw, and linear and rotational velocities of a moving object. Particle filters probabilistically estimate a system's state, $\mathbf{x}$, at time t using the sensor measurement $\mathbf{y}_t$. This is achieved using a set of N particles, each with their own state vector and weight $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$:

$$P(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_t^{(i)} w_t^{(i)} \tag{1}$$

For Gestures Everywhere, we apply the Sequential Importance Sample with Resampling (SISR) algorithm to recursively update the particle filter at each time step [18]. SISR involves updating each particle's state vector using a prediction model: $\mathbf{x}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)})$; weighting each particle by the sensor's likelihood model, given the current sensor measurement and predicted state vector: $w_t^{(i)} = g(\mathbf{y}_t|\mathbf{x}_t^{(i)})$; normalizing the particle weights, and resampling N new particles for $t+1$ using importance sampling [18]. To facilitate the integration of sensor data from several heterogeneous sources, we represent the sensor's likelihood model as a radial basis function:

$$g(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{|\mathbf{y}|} \alpha_j \, \phi(\mathbf{x}, \mathbf{y}_j) \tag{2}$$

where α_j is the weight for the j'th sensor and ϕ is the radial basis function. Figure 2 illustrates the resulting function from fusing three sensors using a Gaussian RBF:

$$\phi(\mathbf{x}, \mathbf{y}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}} \tag{3}$$

Approximating the sensor fusion task as an RBF particle filter has several advantages. First, using an RBF provides an easy method to control how much we should trust each sensor, as this can be directly encoded in α_j, i.e. sensors that provide a better estimation of the true underlying state will have a larger weight. Weights can be set using a-priori knowledge or directly learned from the data using weighted least squares. Second, the RBF mitigates an erroneous sensor (with a low weight) from dominating other sensors that are performing well. Further, the RBF can approximate any multimodal function, such as the functions shown in Figure 2. As particle filters can converge to the true posterior even in non-Gaussian, nonlinear dynamic systems, RBFs provide an excellent compliment to the particle filter. Finally, RBF particle filters are particularly suited for large-scale sensor networks as the fusion algorithm can continue to function even if several of the sensors in a network fail, as the sensor fusion process simply sets $\phi(\cdot)$ for that sensor to zero.

4. ANALYSIS MODULES

In this section, we describe how the RBF particle filters are applied by Gestures Everywhere to aggregate the multi-modal sensor data and estimate the system's spatio-temporal properties. To address each of the spatio-temporal estimations, we define individual particle filters for each low-level property. Depending on the estimation task, each particle filter can represent a local estimate (e.g. the presence estimate at display k), or a global estimate (e.g. the location of a user within the entire building). Given space limitations, we only describe the presence and location modules, however the proximity and identity modules use the same RBF framework.

4.1 Presence Estimation

Presence information can play a key role in improving the design of digital displays. For instance, a system that can sense the presence and approach of people can use that information to reveal possible interactions [15], even at a distance. The GE presence estimation for display k is defined as: $\mathbf{x} = \{p\}$, where p is a real number in the range [0 1] that defines the likelihood of a user being present at display k. The prediction model is set as:

$$p_t^{(i)} = p_{t-1}^{(i)}\gamma + \mathcal{N}_{(0,\sigma)} \tag{4}$$

where γ is empirically set as 0.99 and σ controls how much each particle randomly explores the state space. To compute each particle's weight, features from the RFID reader (y_1), depth camera (y_2), and RGB camera (y_3) at display k are plugged into the RBF to give:

$$w_t^{(i)} = \sum_{j=1}^{3} \alpha_j \, \phi(p_t^{(i)}, y_j) \tag{5}$$

For the RFID reader and depth camera, we directly map the number of users detected by the RFID reader and user segmentation algorithm, scaling these values to a uniform range. For the RGB camera, we extract a movement feature, m, computed from the RGB image I at frame t:

$$m_t = \left[\frac{1}{Z} \sum_i \sum_j |I_t(i,j) - I_{t-1}(i,j)| \right] + \; m_{t-1}\, 0.95 \tag{6}$$

where Z is a normalization constant designed to scale the movement feature to a uniform range, and i and j are pixel indices. Each of these features independently provide good indications of presence, however they all have weaknesses. For example, the RFID and depth features have excellent precision and are robust to false positives, however the depth camera only has a range of approximately five meters and not every user carries an RFID card. Alternatively, the movement feature detects presence if a user is moving, but can fail to detect stationary users and is sensitive to false positives errors. By applying sensor fusion, the strengths and redundancies of each feature can be combined to mitigate each individual sensor's weakness. Table 4.2 shows how the individual accuracy, precision and recall of each of these sensors can be improved using sensor fusion.

4.2 Location Estimation

Location information is often essential for ubiquitous computing systems [12] as it provides an important source of context [20]. The location module estimates the location and orientation of each user within the vicinity of each display in the building. In addition, the location module continually tracks each user for the duration that individual is within the vicinity of a display. A new particle filters is used to track each new user in range of display k, with each particle filter using 500 particles. The particle filter used to estimate each user's location is defined as follows. The state vector is $\mathbf{x} = \{x, y, \theta, \mathbf{c}\}$, where $\{x, y\}$ is the user's estimated location, θ is the user's current heading, and $\mathbf{c}$ is a 10 dimensional vector that represents the user's hue color histogram. The location prediction model is defined as:

$$\theta_t^{(i)} = \theta_{t-1}^{(i)} + \varphi \tag{7}$$

$$x_t^{(i)} = x_{t-1}^{(i)} + \beta \cos(\theta_t^{(i)}) \tag{8}$$

$$y_t^{(i)} = y_{t-1}^{(i)} + \beta \sin(\theta_t^{(i)}) \tag{9}$$

$$\mathbf{c}_t^{(ij)} = \mathbf{c}_{t-1}^{(ij)} + \delta, \qquad 1 \leq j \geq |\mathbf{c}| \tag{10}$$

where φ, β, and δ are Gaussian random variables drawn from $\mathcal{N}_{(0,\sigma_\varphi)}$, $\mathcal{N}_{(0,\sigma_\beta)}$, and $\mathcal{N}_{(0,\sigma_\delta)}$ respectively. σ_φ, σ_β, and σ_δ represent the uncertainty that has built up in the user's x and y location, orientation and color histogram since the previous sample. The sensor's likelihood model is defined as:

$$w_t^{(i)} = \alpha_1 \, \phi(x_t^{(i)}, \hat{x}) + \alpha_2 \, \phi(y_t^{(i)}, \hat{y}) + \alpha_3 \, \phi(\theta_t^{(i)}, \hat{\theta}) + \alpha_4 \, \phi(\mathbf{c}_t^{(i)}, \hat{\mathbf{c}}) \tag{11}$$

where $\hat{x}$, $\hat{y}$, $\hat{\theta}$ and $\hat{\mathbf{c}}$ are the center of mass (COM), orientation and color models of one of the M user's currently visible at display k. $\hat{\theta}$ is computed from a normal vector projected from the user's skeleton torso. Our user segmentation and COM estimation algorithms running on each RGB-D client can robustly detect a new user from the moment they enter the depth field, however, it takes approximately two seconds for the OpenNI/NITE skeleton model to automatically be fitted to each user. The $\hat{\theta}$ feature is therefore not available for the initial detection of a new user. In this case, we take advantage of the RBF model and set $\phi(\theta_t^{(i)}, \hat{\theta})$ to zero until new data is received from the skeleton model. The raw estimate of the user's center of mass already provides a robust estimate of the user's location, however fusing this with the features from the skeleton model and hue color histogram provide an estimate of the user's current orientation in addition to robust tracking of a user while they are in the vicinity of a display. Further, if several depth and RGB cameras overlap at the same location then the raw sensor data from each overlapping sensor can be fused to improve the overall location estimate.

Testing the output of the location module against a one hour recording of hand-labeled data recorded from six digital displays throughout the Media Lab, the tracking algorithm achieved an accuracy of 97% while tracking 51 users in complex scenes with several occlusions. It is important to note that the location module estimates the location of each user within each display's local frame of reference. Consequently, the location modules can efficiently track a user with as little as 500 particles, which is several orders of magnitude less particles required to accurately track a user throughout an entire building [20]. Nevertheless, as we know the location and orientation of each display within the building, we can project the user's location into the building's coordinate system so other tracking systems can take advantage of the fused data, such as those described in [16]. While increasing

Presence Analysis Results	Accuracy	Precision	Recall
RGB Camera Feature	80%	0.77	0.42
Depth Feature	80%	1.0	0.28
RFID Feature	79%	1.0	0.26
RBF Particle Filter	86%	0.99	0.52

Table 1: The results from testing the output of the presence module against one hour of hand-labeled data (~108000 sample points) recorded across six digital displays located throughout the MIT Media Lab. Note that the particle filter achieves better accuracy, precision and recall than any of the individual sensors with just 100 particles.

the number of particles can improve the tracking accuracy, we balance overall accuracy with realtime responsiveness as the system potentially needs to simultaneously track several users at each display.

5. GESTURE RECOGNITION

In addition to the spatio-temporal properties described in section 4, Gestures Everywhere also runs higher-level analysis modules, with the most prominent of these being gesture recognition. Gesture recognition offers many advantages for large-scale public displays, as it enables users to start interacting with the display at a distance or on the move. The gesture recognition module detects if any user at a display performs any of the gestures within the GE gestural dictionary. This consists of a set of fundamental gestural primitives that other ubiquitous applications might want to recognize. Based on prior research [10], we opted to constrain the size of the gestural vocabulary to a small set of upper body gestures, each with high postural disparity from the other gestures to ensure high reliability and recognition rates. This includes a set of core interactions illustrated in Figure 3 that include pointing; next/previous swipes; and a 'home' gesture. These gestures are recognized using the Gesture Recognition Toolkit [8] with a Naïve Bayes classifier [9]. The input to the classifier consists of a twelve-dimensional feature vector, with the x, y, and z values of the user's left and right hands, translated and rotated in relation to the user's torso (making each gesture invariant to the location that the gesture is performed), in addition to the x, y, and z velocities of the user's left and right hands. To mitigate false-positive classification errors, a user's gestures are ignored if the overall likelihood of the skeleton joint model is low, such as when the user is occluded by another individual, or is too close to the depth camera. Client applications can also actively reduce false-positive recognition errors by placing additional contextual constraints on what gestures are actioned. This can include limiting gestures to only be actioned if the individual is stationary, or using the output of the location module to filter any gesture events from users that are not orientated towards a specific target (such as one of the Glass Infrastructure displays).

6. DISCUSSION

Gestures Everywhere has been deployed throughout the Media Lab since early 2013. In this section, we describe some of the real-time pervasive applications that leverage the GE data and discuss a number of key lessons that have been learned throughout the systems deployment.

6.1 Pervasive Applications

Public digital displays can significantly benefit from pervasive analytics as applications can react to whether, which, and how users are interacting with a particular display. The real-time GE analytical data are currently being used to power a number of pervasive applications that run on the digital displays located throughout the Media Lab. For example, using the proximity and identity data we have built an on-the-go update application that displays time-sensitive and location-sensitive information to registered users as they approach the screens. This includes a personal welcome screen to users as they enter the building each morning, or important reminders that can be triggered by both a user's current location or time-of-day. This application can also be used to propagate messages to specific users as they move about the building. As this application may contain sensitive personal information, the on-the-go application will only be shown if the user-identification module reaches a specific confidence threshold; which the user can control via their personal data-portal on the GE website. This enables users to control the privacy settings of the system to best match their individual needs. In addition to on-the-go, we have integrated the real-time location data into the Glass Infrastructure interactive map. The interactive map visualizes the location of anonymized users as they walk throughout the building and facilitates users to directly search for another individuals' current location within the map (see Figure 1).

6.2 Scaling & System Modularity

Building flexible pervasive systems that can scale to an entire building can be challenging. Our system has currently scaled to include approximately 100 sensors distributed across a 100K sq. ft. six-floor building, with a database of 1800 identifiable users. This is managed by one server which runs all the data collection, sensor fusion, analysis modules and API interfaces. A core design of the GE architecture that has facilitated the system to easily scale to an entire building is that much of the processor-intensive computer vision and machine learning algorithms are run locally on each of the devices managing the Glass Infrastructure displays. The result of this architecture is that the remote RGB-D clients send pre-computed features to the GE server. This significantly reduces the overall-processing overhead and amount of real-time data being streamed to the server. A second important design of the system is its abstract modularity. Each of the system's components, such as remote sensor interfaces, spatio-temporal analysis modules, gesture recognition, etc., are separated by robust software abstractions. These abstractions facilitate new sensors to easily be added to the system as they are added to the environment; pop-up displays to rapidly be integrated into the system for a one-day-event; or additional analysis modules to be added that build on the output of the existing modules. Further, our system exposes the data from all of these modules via two accessible APIs in HTTP/JSON and OSC. This enables multiple pervasive applications to access the raw, fused, and analytical data; facilitating the GE data to be used as input to other pervasive systems that may be added in the future to the building's digital ecosystem. By separating the remote sensor interfaces from the pervasive analytics, and abstracting the output of the system from the pervasive applications that run on top of it, we provide a robust, flexible infrastructure for dynamically connecting multiple sensing

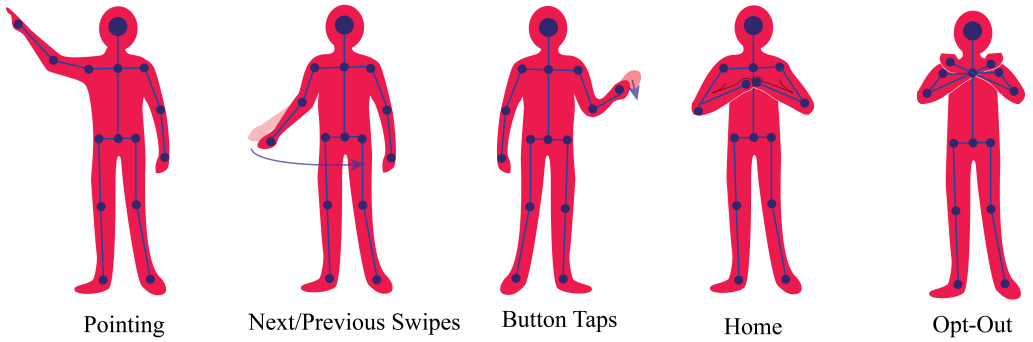

Figure 3: The core set of gestures recognized by the gesture recognition modules. The pointing, swipe and tap gestures can all be performed by either the left or right hand.

technologies to multiple applications.

6.3 Privacy

Privacy is clearly an important issue in the development of any pervasive system. To protect an individuals' privacy, personally identifiable data is only released through the GE system if the user in question allows this. A central design-philosophy of the system is that users have full control over their own data and can choose to share their personal tracking data with a specific group of individuals; users currently inside the building; everyone; or no one at all. Users can easily control their sharing settings at anytime via the Glass Infrastructure displays or on their personal devices via a web browser. In addition to controlling how other users view an individuals data, each registered user has full access and ownership of their own data. Registered users can login to the GE system via their personal data-portal and browse their data repository; facilitating individuals to understand just how visible they are to the system, who they are sharing their data with and monitor who has been searching for them.

7. CONCLUSION

In this paper, we have presented Gestures Everywhere, a dynamic framework for multimodal sensor fusion, pervasive analytics and gesture recognition. We have described how our system applies radial basis function particle filters to maximize the heterogeneous sensor data deployed on a network of digital displays, and demonstrated the integration of these algorithms into a scaleable system that supports a network of interactive displays and the pervasive devices within their vicinity.

8. REFERENCES

[1] J. K. Aggarwal and N. Nandhakumar. Sensor fusion and aerospace applications. In *Sensor Fusion and Aerospace Applications*, volume 1956, 1993.

[2] K. Bernardin and R. Stiefelhagen. Audio-visual multi-person tracking and identification for smart environments. In *Proceedings of the 15th international conference on Multimedia*, pages 661–670. ACM, 2007.

[3] M. Bletsas. The mit media lab's glass infrastructure: An interactive information system. *Pervasive Computing, IEEE*, 11(2):46–49, 2012.

[4] A. Brooks and S. Williams. Tracking people with networks of heterogeneous sensors. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2003.

[5] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pages 176–183, 2000.

[6] D. Fox, D. Schulz, G. Borriello, J. Hightower, and L. Liao. Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3):24–33, 2003.

[7] T. Germa, F. Lerasle, N. Ouadah, and V. Cadenat. Vision and rfid data fusion for tracking people in crowds by a mobile robot. *Computer Vision and Image Understanding*, 114(6):641 – 651, 2010.

[8] N. Gillian. Gesture Recognition Toolkit. `http://www.nickgillian.com/software/grt`.

[9] N. Gillian, R. B. Knapp, and S. O'Modhrain. An adaptive classification algorithm for semiotic musical gestures. In *the 8th Sound and Music Computing Conference*, 2011.

[10] K. Grace, R. Wasinger, C. Ackad, A. Collins, O. Dawson, R. Gluga, J. Kay, and M. Tomitsch. Conveying interactivity at an interactive public information display. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, pages 19–24. ACM, 2013.

[11] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *UbiComp 2004: Ubiquitous Computing*, pages 88–106.

[12] J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. In *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*, pages 22–28. IEEE, 2002.

[13] M. Laibowitz, N.-W. Gong, and J. A. Paradiso. Wearable sensing for dynamic management of dense ubiquitous media. IEEE, 2009.

[14] J. Madapura and B. Li. Multi-target tracking based on kld mixture particle filter with radial basis function support. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 725–728.

[15] N. Marquardt and S. Greenberg. Informing the design of proxemic interactions. *IEEE Pervasive Computing*, 11(2):14–23, 2012.

[16] S. Pfenninger. A people tracking system utilizing particle filters in a non-overlapping sensor network. Master's thesis, Swiss Federal Institute of Technology (ETH), 2013.

[17] T. Teixeira, D. Jung, and A. Savvides. Tasking networked cctv cameras and mobile phones to identify and localize multiple people. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 213–222. ACM, 2010.

[18] S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. *Intelligent robotics and autonomous agents, The MIT Press (August 2005)*, 2005.

[19] X. Wang, S. Wang, and J.-J. Ma. An improved particle filter for target tracking in sensor systems. *Sensors*, 7(1):144–156, 2007.

[20] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123. ACM, 2008.

[21] M. Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(03):193–200, 2005.