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

Nicholas Gillian¹, Sara Pfenninger², Spencer Russell¹, and Joseph A. Paradiso¹
{ngillian, saras, sfr, joep}@media.mit.edu

¹Responsive Environments Group
Massachusetts Institute of Technology Media Lab, Cambridge, MA, USA
²Wearable 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]. Consequently, 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, scalable, 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
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. 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 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. 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
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 $y_t$. This is achieved using a set of $N$ particles, each with their own state vector and weight $\{x_{t,i}, w_{t,i}\}$:

$$P(x_{t+1}, y_{1:t+1}) \approx \frac{1}{N} \sum_{i=1}^{N} x_{t,i} w_{t,i}$$

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: $x_{t+1}^{(i)} = f(x_{t}^{(i)});$ weighting each particle by the sensor’s likelihood model, given the current sensor measurement and predicted state vector: $w_{t+1}^{(i)} = g(y_{t}, x_{t}^{(i)});$ 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:

$$g(y|x) = \sum_{j=1}^{k} \alpha_j \phi(x, y_j)$$

where $\alpha_j$ is the weight for the $j$th sensor and $\phi$ is the radial basis function. Figure 2 illustrates the resulting function from fusing three sensors using a Gaussian RBF:

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

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 $\alpha_j$, 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 multimo
dal sensor data and estimate the system’s spatio-temporal
properties. To address each of the spatio-temporal estima-
tions, 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 esti-
mate 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, how-
ever 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 in-
formation to reveal possible interactions [15], even at a dis-
tance. The GE presence estimation for display k is defined
as: $x = \{x, y, \theta, c\}$, where $x$ is the user’s esti-
mated location, $\theta$ is the user’s current heading, and $c$
is a 10 dimensional vector that represents the user’s hue color
histogram. The location prediction model is set as:

$$\hat{\theta}^{(i)} = \theta_{k-1}^{(i)} + \varphi$$

$$x_k^{(i)} = x_k^{(i)} + \beta \cos(\hat{\theta}^{(i)})$$

$$y_k^{(i)} = y_k^{(i)} + \beta \sin(\hat{\theta}^{(i)})$$

$$c_k^{(i)} = c_k^{(i)} + \delta_i, \quad 1 \leq j \leq |c|$$

where $\varphi$, $\beta$, and $\delta$ are Gaussian random variables drawn
from $N(0,\sigma_{\varphi}), N(0,\sigma_{\beta}),$ and $N(0,\sigma_{\delta})$ respectively. $\sigma_{\varphi}, \sigma_{\beta},$ and $\sigma_{\delta}$ 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:

$$w_k^{(i)} = \alpha_1 \phi(x_k^{(i)}, \hat{x}) + \alpha_2 \phi(y_k^{(i)}, \hat{y}) + \alpha_3 \phi(\hat{\theta}^{(i)}, \hat{\theta}) + \alpha_4 \phi(c_k^{(i)}, \hat{c})$$

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 $t$:

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

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
cannot 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 miti-
gate 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 com-
puting 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 dis-
play in the building. In addition, the location module con-
tinually 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 $x = \{x, y, \theta, c\}$, where $\{x, y\}$ is the user’s es-
timated location, $\theta$ is the user’s current heading, and $c$
is a 10 dimensional vector that represents the user’s hue color
histogram. The location prediction model is defined as:

$$\hat{\theta}^{(i)} = \theta_{k-1}^{(i)} + \varphi$$

$$x_k^{(i)} = x_k^{(i)} + \beta \cos(\hat{\theta}^{(i)})$$

$$y_k^{(i)} = y_k^{(i)} + \beta \sin(\hat{\theta}^{(i)})$$

$$c_k^{(i)} = c_k^{(i)} + \delta_i, \quad 1 \leq j \leq |c|$$

where $\varphi$, $\beta$, and $\delta$ are Gaussian random variables drawn
from $N(0,\sigma_{\varphi}), N(0,\sigma_{\beta}),$ and $N(0,\sigma_{\delta})$ respectively. $\sigma_{\varphi}, \sigma_{\beta},$ and $\sigma_{\delta}$ 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:

$$w_k^{(i)} = \alpha_1 \phi(x_k^{(i)}, \hat{x}) + \alpha_2 \phi(y_k^{(i)}, \hat{y}) + \alpha_3 \phi(\hat{\theta}^{(i)}, \hat{\theta}) + \alpha_4 \phi(c_k^{(i)}, \hat{c})$$

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 com-
plex 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 lit-
tle as 500 particles, which is several orders of magnitude
less particles required to accurately track a user through-
out an entire building [20]. Nevertheless, as we know the loca-
tion 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
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.

<table>
<thead>
<tr>
<th>Presence Analysis Results</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB Camera Feature</td>
<td>80%</td>
<td>0.77</td>
<td>0.42</td>
</tr>
<tr>
<td>Depth Feature</td>
<td>80%</td>
<td>1.0</td>
<td>0.28</td>
</tr>
<tr>
<td>RFID Feature</td>
<td>79%</td>
<td>1.0</td>
<td>0.26</td>
</tr>
<tr>
<td>RBF Particle Filter</td>
<td>86%</td>
<td>0.99</td>
<td>0.52</td>
</tr>
</tbody>
</table>

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
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