Event-Driven Low-Power Gesture Recognition Using Differential Capacitance

Gurashish Singh, Member, IEEE, Alexander Nelson, Student Member, IEEE, Sheung Lu, Student Member, IEEE, Ryan Robucci, Member, IEEE, Chintan Patel, Member, IEEE, and Nilanjan Banerjee

Abstract—Individuals with mobility impairments represent a large portion of the population. These impairments are often the sequel of strokes, spinal cord injury, or diseases, such as amyotrophic lateral sclerosis and Guillain–Barré syndrome. Many of these diagnoses manifest in upper-extremity mobility impairment, which may prohibit or make difficult actuating devices within the home. While solutions exist for smart-home automation and control, a few are approachable by those with mobility impairments. To address this issue, we designed Inviz, a touchless low-power assistive gesture recognition system that utilizes an array of textile capacitive sensors to control a smart-home automation system. We present two novel research contributions; the use of flexible textile-based capacitive arrays as differential proximity sensors, and a low-power event-driven hierarchical signal processing algorithm for capacitor sensor arrays. A fully functional prototype system is implemented and evaluated in the context of an end-to-end home automation system using intuitive swipe and hover gestures that are comfortable and approachable for users with mobility impairments. The evaluation demonstrates that the system achieves high accuracy (>90%) while maintaining low latency (<1 s) and low energy consumption (2475 µW). In addition, we evaluate the system on a subject with upper extremity mobility impairment to verify its usage as an accessibility device.

Index Terms—Capacitors, sensor arrays, assistive technology, assistive devices, gesture recognition, wearable sensors.

I. INTRODUCTION

SMART home systems and cyber-physical interfaces have permeated the technological landscape in recent years. Despite this uptick in bridging the physical world to a more universally accessible sphere, there remain very few interfaces which consider people with upper-extremity (UE) mobility impairment. This population represents a large portion of society. In fact an estimated 1.5 million individuals in the United States alone are admitted to hospitals each year because of injuries (spinal-cord, traumatic brain) and stroke [1]–[3].

Manuscript received December 25, 2015; accepted January 31, 2016. Date of publication February 16, 2016; date of current version May 17, 2016. This work was supported in part by the National Science Foundation under Grant CNS-1305099 and Grant CNS-1308723, and in part by the National Science Board under Grant IIS-1406626. This is an expanded paper from the IEEE Pervasive Computing and Communications 2016 Conference. The associate editor coordinating the review of this paper and approving it for publication was Prof. Ravinder S. Dahiya. (Gurashish Singh and Alexander Nelson contributed equally to this work.)

The authors are with the Department of Computer Science and Electrical Engineering, University of Maryland at Baltimore County, Baltimore, MD 21250 USA (e-mail: singh1@umbc.edu; alnel1@umbc.edu; sb2@umbc.edu; robucci@umbc.edu; cpate2@umbc.edu; nilanby@umbc.edu). Digital Object Identifier 10.1109/JSEN.2016.2530805

These diagnoses can oft result in weakness, paresis, or paralysis to a broad region because of damage to the nervous system. Changes to healthcare has resulted in shorter length stays even though these circumstances often require extensive rehabilitation to achieve meaningful physical recovery [4], [5]. Assistive technologies which interact with the user to maximize independence early and efficiently can ease the physical and economic burden which result from this transitory period. For injuries which have long-term or permanent effect, integrated assistive technologies which are embedded into the individuals environment can create an accessible living space capable of allowing autonomy within their own home.

Gesture recognition is fast becoming a common method for device control, and is approachable by individuals with UE motor disabilities. A 2014 study of interviews with individuals who have motor impairments concludes that an intuitive, reliable, easy to set up system is important for continuance of use [6]. Gesture recognition has been performed in the past using inertial sensors, vision systems, resistor sensing, and other approaches which capture body motions [7]–[12]. UE mobility impairments, however, present fundamental challenges which the above approaches often fail to address. Vision-based systems, such as eye-tracking or body gesture, are typically rigidly fixed, having a single orientation in which they will recognize a gesture. If the camera is in the environment, then the user’s body may occlude the system. Cameras fixed to the user are obtrusive, calling unnecessary attention to the disability which leads often to abandonment [13]. Physical contact systems such as tactile switches and resistive sensing devices can cause skin contact injuries. Similarly, evoked-potential systems which use electrodes and often an electrolytic gel can cause skin irritation and abrasion. These skin injuries create conditions that can have a deleterious effect if unnoticed due to diminished sensation in the extremities, which is common for individuals with UE disabilities. Moreover, existing solutions are often not suitable for those with impairments as they assume certain motions which a person may not be able to complete.

Capacitive-Sensor Arrays (CSA) utilize a change in capacitance to detect remote bodies. Using differential capacitance on a fixed geometry, movement gradients can be determined by the combinatorial change in capacitance between pairs of capacitors. These sensors are made from metal-infused textiles (thread and cloth) that can worn or otherwise integrated easily into the environment of persons with UE mobility impairments (e.g. wheelchairs, bed-sheets) rendering the sensing modality...
effectively invisible. We have implemented and evaluated textile CSAs in a fixed geometry as an assistive gesture recognition system for UE motor impairment. Our solution, Inviz, is responsive and accurate as demonstrated by our evaluation. Moreover, we integrate the sensors into clothing material, and perform the gesture recognition on a small embedded controller such that the entirety of the system is minimally obtrusive, a major deterrent of long-term use [6]. The fabric sensors are subject to the same wear-and-tear as traditional fabrics, but are themselves inexpensive and replaceable, as the computation module is detachable and reusable. Figure 1 demonstrates the prototype system which we have developed using fabric capacitive plates and conductive threads sewn into denim to emulate a system built into jeans. The sensors are sensitive to small movements within a reasonable range (<10 cm) of the sensors. The data from the sensors is collected and analyzed using a low-power hierarchical signal processing algorithm which converts the differential capacitive signals into gestures. We have connected this to a Smart-Home automation system to control appliances. The system supports two types of gestures: a “Swipe” which involves moving the hand from one area to another, and a “Hover” which involves placing the hand above a single area, and then removing it from the array. These gesture types were established as typically approachable and comfortable for people with UE motor impairments in conversations with medical professionals (e.g. Physical Therapists) and individuals with partial paralysis (though capabilities within this population vary widely depending on diagnosis, therapy, and the individual). The use of CSAs allows for non-contact gestures, which prevents skin-contact injuries, and imprecise gestures which can be common in those with tremor or extended weakness.

II. RELATED WORK

The system and the ideas presented within build upon the body of work in capacitive sensing, gesture recognition, and signal processing for gesture recognition. Our work is compared with relevant literature below.

A. Capacitive Sensing

Our system utilizes and builds on knowledge from capacitive sensing [14] applied to many areas including industrial, automotive, and healthcare applications [15], [16]. Additionally, these sensors have been used for positioning [17], [18]; humidity sensing [19], [20]; tilt sensing [21], [22]; pressure sensing [23]; and MEMS-based sensing [23]. Capacitor sensors have been extended to proximity sensing applications for robotics, industrial monitoring, and healthcare [24]–[26]. More recently, products such as Microchip’s GestIC [27] have been developed which allow for 3D touchless gesture tracking through the use of a rigid array of capacitor sensors. Our system extends the use of capacitor sensors in two distinct fashions. First, the use of textiles as a capacitor sensor array embedded into objects of everyday use for remote gesture sensing is novel within the field to the best of our knowledge. This particular application faces several unique challenges, especially as applied to users with UE mobility impairments. That the sensors are not rigid increases uncertainty within the system. Further, Inviz utilizes an intelligent hierarchical signal processing algorithm for ambient gesture recognition within the context of flexible capacitor sensors, allowing for a responsive and low-power solution. We have outlined the approach which allows for customization both on the recognition scheme and in terms of personalization to the user space.

B. Gesture Recognition Systems

Gesture recognition systems have emerged as an intuitive and easily approachable solution for environmental control. Many such systems utilize vision for gesture recognition [28], which allow environmental control without necessitating contact with a physical device. However, vision based systems either require blanket coverage of the home, which can cost thousands of dollars, or a user-mounted camera. The development of assistive home-automation solutions for persons with disabilities has been adopted to enable greater autonomy in terms of environmental control. Examples of such systems include voice-activated systems [29], head and eye tracking systems [30], and inertial sensor based systems [12]. The ability to customize these systems is imperative, as many diagnoses can express in a wide range of severity. For UE, the system should be able to function in any space; it should be responsive and approachable, regardless of orientation or position of the user. Inviz allows for subtle arbitrary gestures within the user’s own reference frame as it is built into the environment of the user, notably into clothes, bed sheets, or as a pad.

C. Signal Processing for Gesture Recognition

Signal processing as applied to gesture recognition is an established field with a large body of work. Hidden Markov
Models [31], Bayesian inference [32], and decision trees [33] have all been applied to gesture recognition to convert data from sensors into movements and gestures. Our system utilizes this higher-level learning as a classification tool. The innovation within Inviz is the combination of feature extraction and learning techniques within these different power tiers to responsively process flexible CSA data with minimal energy consumption.

D. Previous Work

We extend on our own work in this area through analysis and characterization of the fabric capacitor sensors, the analysis of the system as it applies to individuals with limited mobility and with the addition of virtualized combinatorial differential sensors [34]. A separate parallel work utilized a similar sensing platform with multiple single-ended measurements of capacitance for detailed position-tracking and classification system for recognition of complex drawn gestures [35]. In that work, training and test gesture data was preprocessed to ignore position, scale, and rotation in order to improve classification performance for prescribed complex gestures. The newly presented work herein instead exploits position, scale and rotation features in the design of sensing and processing to provide a simpler gesture interface especially useful to some with injuries that co-occur with traumatic brain injury. The sensors themselves are differentiated by the utilization of geometric differential connections and the sensing modality and processing scheme uses event-driven differential capacitive sensing and recognition.

III. Capacitive Sensor Arrays for Gesture Recognition

In order to address the holes left by existing assistive systems in regards to touch, precision, occlusion, and obtrusion, we implemented a wearable, effectively invisible, sensing platform through the integration of textile capacitive sensors. While many capacitive sensor applications utilize touch-based sensing, we utilize capacitive sensing as remote-body detection as a touchless gesture recognition system.

The capacitor sensors can be touchless because of the mode in which the sensors can detect changes in the electric field. The use of sensor plates separated by air as the medium creates a capacitor where the capacitance is relative to the distance between plates and the dielectric, which in the nominal case is air. When the user places their hand into the area around the plates, the perturbation of the electric field is detected by a change in the effective dielectric. This method of sensing allows for versatile remote movement detection as compared to inertial sensors which measure only movement relative to the body of which they are attached. Figure 2 is a representation of the capacitor sensors being worn by a user who is performing gestures above the array. As the user moves their hand close to the array, the capacitance $C_b$ increases, and therefore the inverse of $C_b$ can be used to localize the distance of the hand with respect to a single plate. Figure 2 also demonstrates the cross-section of the sensor array, which is composed of a grid of fabric capacitive sensors connected via conductive threads.

We use a shielding plane to restrict capacitance measurement to above the array and to minimize parasitic capacitance and noise coupling. The last layer is a ground plane which couples the human body to the ground of the sensor which provides a common reference for the capacitance measurements.

A. Sensor Characterization

A textile capacitor plate can be fabricated using two methods. The first method uses fabric capacitor plates (see Figure 3 (a) right) that are cut from metal infused fabric sheets and ironed and sewn into their proper positions on some fabric (e.g. denim). Routes are drawn to the plates using metal infused threads sewn into the fabric and attached to the sensor board using vampire clip adapters. Alternatively, one may use embroidery (see Figure 3 (a) left) to create
these plates from metal-infused threads, thereby automating
the process of creating Inviz sensors using CNC sewing and
embroidery machines. We characterize the range of both
types of capacitor plates in Figure 3 (b). In the experiment, we
fabricate circular capacitor plates of varying radius and then
measure the capacitance as the distance between a user’s hand
and the plate is varied. We experimented with three capacitor
plate diameters (35mm, 45mm, 60mm). From Figure 3 (b) we
can draw two conclusions. First, we find that both method of
fabricating the sensors have similar exponential capacitance-
distance characteristics. However, the embroidered sensor
plates have lower overall capacitance. This is due to the density
of the metal in the embroidered plates which is less than
the fabric plates. A larger capacitance can be achieved by
increasing the size of the plate or by creating thicker weaves.
Second, we find that the smallest plates have a range of close
to 4 inches after which the capacitance does not change with
distance between the hand and the plate. These values are
consistent with those found by Rus et al. [36] who perform an
analysis of capacitor sensor sensitivity in multiple forms. This
range is sufficient to prevent accidental touch and skin abrasion
and can be adapted as the range is variable with plate size and
shape. We have developed two sensors with 4 and 12 sensor
plates respectively using textile fabric plates (see Figure 1).

B. Capacitive Sensor Arrays

This system uses an array of textile capacitive sensor plates,
granting multiple advantages compared to a single plate
system. First, the use of differential capacitive measurements
aids in minimizing the noise due to stray capacitance and
movements within the array. Additionally, the geometric
properties of the array in combination with the differential
capacitance measurements can be used to extract rich motion
artifacts such as velocity, which can be useful in the determina-
tion of different gestures. Figure 4(a) is a demonstration of
one of these features. The figure displays the analog difference
between two capacitor plates as a person performs a “swipe”
gesture and moves their hand from one plate to another. The
differential capacitance graphed is shown as a peak, followed
by a zero crossing, followed by a valley. The peak-to-valley
time can be utilized as the inverse of the velocity in the
direction specified by the angle of the sensor-pair, with the
direction of the vector specified by the relative ordering of
the peak-valley. Figure 4(a) also graphs the analog difference
of a “hover” gesture, demonstrating the differential capacitance
between plates when the user maintains their hand above a
single plate. In this case, the width of the peak corresponds
to the time in which the “hover” gesture took place.

The examples above serve as demonstrations as to how an
array of capacitors can be used to capture high-level movement
features such as time, velocity, and relative position of a
foreign body with respect to the array. The use of an array
of these sensors and their combinatorial sensor pairs provide
multiple vantage points of the each movement and can increase
the reliability of gesture recognition. Our system converts data
from the CSA into reliable gestures through a hierarchical
signal processing algorithm which we detail in Section IV.

The intuition behind some of the signal processing is shown
in Figure 4(b), which illustrates one of the fundamental
challenges in processing the raw differential capacitance data.
Pictured in the figure is the same gesture which is performed
three times by the same user, with regions i and ii demonstrat-
ing the variability and irregularity of the raw capacitance.

IV. HIERARCHICAL SIGNAL PROCESSING

We utilize a hierarchical design to distribute the capaci-
tive data processing into several computational tiers, includ-
ing observation, feature extraction, and machine learning as
depicted in Figure 5(a). The observation tier itself utilizes
hardware invoked low and high power tiers to remain sensitive
when a user is approaching the array, but in a low power
tier while the array is in disuse. The data flow moves from low-power special-purpose hardware to a general purpose micro-controller which is woken up as-needed when pertinent features are collected. Figure 5(b) illustrates the conceptual blocks for transforming raw data from the capacitor plates into gestures. This design allows the system to be responsive and accurate while using a minimal amount of energy. The details of each of the different processing tiers in our hierarchy are illustrated in Figure 5(b).

Though our evaluation utilizes a gesture set which is composed of only “hover” and “swipe” type gestures, the signal processing for the system is intended to be a general framework through which arbitrary gestures can be recognized from an array of wearable fabric capacitors while consuming very little energy (<3 mW). This architecture can potentially be extended to support a more complex gesture set.

A. Observation Calculation

The calculation of the input observations occurs at the lowest tier of the hierarchy and are calculated as linear combinations of the capacitance from each of the sensors of the array. The observations \(\{y_1, ..., y_k\}\), are created through the observation model \(\sum_{i=1}^{n} W_{i,k} \cdot c_i\) where \(W_{i,k} \in \{0, 1, -1\}\), \(c_i\) represents the analog capacitance between a plate and the ground, and \(n\) is the total number of capacitor sensors (4 for our system). Our system utilizes a single capacitance to digital conversion unit, and therefore the observation vector is not taken concurrently. Instead, the measurements are recorded periodically in a sequential pattern which compose a single observation vector \(\{y_1, ..., y_k\}\). The linear combinations provided to the capacitance-to-digital converter for each observation are controlled through the use of low-power analog multiplexers. The exact ordering of the measurements, in fact, does not have an effect on recognition provided that gesture motions are slow compared to the sampling rate. The impact of the ordering is further diminished in our hierarchy through the use of machine learning at the top computational level. Much of the transient environmental noise is rejected through the use of analog differential measurements, including any common noise from among the plates. Differential measurements trade off some amount of potential measurement distance for accuracy as the differential measurements form a receptive field which is most sensitive to motions in the proximity of the plates, but is more insensitive to motions at a distance as compared to a single-ended measurement. This is acceptable, and maybe preferable as the differential measurements can cancel noise due to stray movements which are far from the sensor array, but instead can capture more subtle movements closer to the array.

In Figure 4(a-i) we demonstrate a characteristic response from the differential capacitor sensors during a “swipe” gesture. This gesture has the hand passing successively over the two sensors. Figure 4(a-iii) similarly shows the analog differential during a “hover” gesture in which the hand only passes above a single sensor. These two responses are captured in the system utilizing a pair of thresholds and detectors which capture events as illustrated in Figure 4 as the high (green) and low (red) thresholds. Figure 4(a-ii and a-iv) demonstrate what is captured in the system as two binary responses, green showing when the signal is above the high threshold and red when the signal is below the low threshold. Event thresholds are established comparative to a baseline measurement for each observation channel. \(\{y_1, ..., y_k\}\), and is continually recalculated when there is no activity near the sensor (i.e. when there is a minimal change in capacitance \(\Delta C\)). Experimentation was used to determine the minimum separation of the thresholds from the baseline, but is dependent upon the size of the plates that make up the CSA. The binary threshold-event values are collected using an ultra-low power capacitance-to-digital conversion IC which provides threshold-crossing detection and manual offsets. Threshold events are often used to determine when a user has touched a capacitor sensor, but we have exploited the measurement to drive the event generation for the higher-tier calculations. This requires a much lower and noise-
prone threshold which must be accounted for in higher tiers. Further, irregularity of the sensors (including potential degradation of signal quality with wear) and irregularity between similar gestures necessitate a more sophisticated higher-tier processing than what is required for simple touch-based gestures. Additionally, the CSA measures more complex signals due to the combination of the user’s hand, forearm, and wrist, which is not present when using a single-point finger touch. Below, we discuss how the binary outputs coming from the digital threshold detectors are used to build higher-level features which are passed into the final stages of machine learning-based classification.

B. Event Message Generators

The next level of our processing hierarchy utilize thresholding signals for two important purposes. The binary signals themselves are the only representation of the actual signal that is passed into the event detection. This represents a compact representation of the signal, which minimizes the memory requirements of the system for capturing the signal history. This simplicity also enables real-time recognition when using higher-level feature-extraction and machine learning. Additionally, the binary signals serve as interrupt signals which wake-up the higher-level processor. This wake-up threshold, determined to be approximately 12 cm, was determined experimentally to be sensitive in our laboratory experiments, though experimentation during in-situ trials would be needed to determine the relevance of this threshold in terms of false-positive and false-negative rates. For each of the observations \( y_k \) there is a defined upper and lower threshold, \( TU_k \) and \( TL_k \) respectively, which are calculated as an offset from the baseline which is defined by the measurement IC. Based on these thresholds and binary inputs, we define two signals, \( BP_k \) and \( BN_k \) which are the positive-peak and negative-peak binary signals respectively that are collected by the processor by the following formula:

\[
BP_k[n] = \begin{cases} 
  \text{TRUE} & \text{if } y_k[n] > TU_k \\
  \text{FALSE} & \text{otherwise}
\end{cases} \quad (1)
\]

\[
BN_k[n] = \begin{cases} 
  \text{TRUE} & \text{if } y_k[n] < TL_k \\
  \text{FALSE} & \text{otherwise}
\end{cases} \quad (2)
\]

where \( n \) is the sample number. Event detectors utilize these binary threshold signals to determine characteristics which are extracted as event features to form an “event message.”

An event is defined in our system as the period of continuous TRUE values for either \( BP_k \) or \( BN_k \) for a given observation channel. Three event features are created for each event, and are defined as: (1) arrival time: the number of samples from the first threshold crossing to the beginning of the event; (2) duration: the amount of samples that the particular binary signal is TRUE; (3) event polarity: a binary variable which determines which of \( BP_k \) or \( BN_k \) is TRUE. At the end of each event, an indicator is sent to the higher-level stage to prepare to process the events.

Gesture recognition is processed either on-line or in post-processing. On-line recognition requires a considerable amount of processing as the classification is performed and updated during the gesture until it reaches a confidence level. This is intractable for a low-power embedded processor. For post-processed gestures, one must determine the end of a gesture, which is an important tradeoff. Ending the gesture early could result in a missed event which could create a misclassification. On the other hand, latency and power consumption of the system are relative to the length of the gesture. We maintain a countdown timer which is incremented by the duration of events and terminate the gesture when the timer reaches zero.

The event duration is indicative of two parameters: (1) the speed of a gesture, and (2) when events could be co-located across multiple observation channels. The intuition behind Algorithm 1, is to prevent a timeout while the gesture may still be in progress by extending the timeout proportional to the length of these events. Once all events are collected, they are propagated to the aggregation and filtering module.

### Algorithm 1 CaptureEventMessages \((\Delta T = \frac{1}{\text{sampling rate}})\)

- **Event** \( E_1 \) = first event.
- **gesturetimeout** = eventduration \((E_1)\)
- for every \( \Delta T \) seconds
  - IF **gesturetimeout** == 0
    - break
  - end IF
  - decrement **gesturetimeout**
  - IF Event \( E_i \) is collected from observation \( i \).
    - **gesturetimeout** + = eventduration \((E_i)\)
  - end IF
- end for
- return TRUE

C. Event Filtering

During evaluation of our prototype, several instances have occurred in which spurious or atypical events are created either due to irregularity in the way gestures are performed or with the sensor data itself.

Figure 5(b) gives examples of these spurious events which can be a product of movement irregularity, and must be filtered using per-channel filtering rules. There are four per-channel rules, defined as: (1) Any event which has a much larger duration than all previous durations in a given observation channel will remove all previous event messages from that channel queue; (2) When there exist two events with the same polarity within a single channel, only the longer of the two events is kept; (3) If consecutive events arrive which have a sizable delay in arrival time compared to an earlier event, the earlier message is removed from the channel queue; (4) If any two event messages have a large difference in their duration, then the shorter message is removed from the channel queue.

After the per-channel filtering, the events are aggregated across all observation channels to apply another set of cross-channel filtering rules. An example of these rules is shown in Figure 5(b). The rules for cross-channel filtering are defined...
as: (1) If any channel's total duration of messages is considerably shorter than the average total, then it is assumed that this channel is receiving noise, and is removed from the event set; (2) If the sum of all durations across the observation channels is small, then the entire gesture is filtered out and reported as a null gesture. After filtering, each observation channel will have either no events, a single event or a pair of complimentary events. If there are no events, then the channel is labeled as NAE for not-an-event. If there is a single event, then the channel is labeled by the polarity of the event (P for positive, N for negative). Finally, if a channel contains a complimentary pair of events, i.e. a positive then a negative event, then the message is labeled as PN for if the channel had a positive followed by a negative event, or NP for the opposite case. In the case of a PN or NP channel, the events are combined, and have a single duration which is calculated as the sum of the pair of durations. In the case of NAE, the duration and arrival time are both zero.

D. Event Characteristics

The features calculated from all of the events is then passed to the gesture classifier. Each observation channel reports the following information, called “Event Characteristics”: Event Label (P, N, PN, NP, NAE), arrival time with respect to first event, and duration. Each gesture contains a unique set of event signatures which are important for distinguishing between gestures. For instance, the combination of the duration and arrival time encode the velocity of the gesture (both speed and direction of motion), which can be used to distinguish gestures, and potentially between separate users. The combination of all of the features are used by the machine learning algorithm to infer the gesture classification. All signal processing occurs in real-time on the micro-controller.

E. Machine Learning Algorithm

The highest level of our signal processing hierarchy is a supervised machine learning classification. The classifier takes in the filtered message bundles as its observation model, and gives as output the classification. To obtain the supervised training data, each subject is asked to perform each of the gestures from the set a number of times, and that is inserted into the training data set with its correct labeled gesture. To determine the classifier which we would use for the real-time system, we experimented with several low-level classifiers including Nearest Neighbor, Decision Trees, and Naive Bayes. The relative accuracies and complexity trade-offs are compared in Section VI. When the classifier has finished and determines the gesture output, a Bluetooth Low Energy (BLE) module is woken and transmits the gesture type to a base-station which is used to control the appliances of a custom smart-home automation system.

V. Prototype Implementation

For the development and evaluation of this system, we have implemented an end-to-end fully functional prototype cyber-physical system for home automation, which is illustrated in Figure 1. The classified gestures are transmitted over a BLE connection to a personal computer which controls the smart-home appliances over a Z-Wave connection through the Micasaverde Vera gateway. Our prototype is composed of a custom-designed PCB which is stocked with: a low-power capacitance measurement IC which can perform the measurement, observation calculation, and thresholding; an MSP430 micro-controller, and a BLE module for wireless communication. The system is powered by a 1 Ah Li-Ion rechargeable battery. The CSA is created from metal-infused fabric which is sewn into denim, and routed to the measurement circuit with 4-ply conductive thread which has a linear resistance of approximately 50 Ω/m.

During this implementation we faced two specific design challenges for the integration of textile-based wearable capacitor sensors. First, the threads used for the routing are created by weaving silver-plates/spun-stainless steel threads together with non-conductive thread. This composition, however, has a tendency to fray at the ends, which can cause microscopic shorts between capacitor plates of adjacent threads. This problem is particularly difficult to diagnose, as the two sensors may appear to be functioning because their sensor values are a combination of the two plates. This is mitigated by maintaining enough space between threads, and coating the connections to prevent fraying. Connecting the threads to the measurement circuit is also difficult as the threads combust with too much heat, and therefore are resistant to soldering. We mitigated this problem through the use of vampire tap connectors which clip into the thread.

For the sensor array, we used two different array configurations, one with four sensor plates and the other with twelve sensor plates as shown in Figure 1. Increasing the number of capacitor plates on the sensor allows us to explore the fidelity of gesture recognition as the spatial resolution of the sensors is increased. The use of the four plate sensor is straightforward – each physical sensor plate is considered as the sole component of a positive or negative terminal, with directionality established within the two-dimensional plane. The 12-plate sensor was constructed to maximize the capability of our measurement IC. In the 12-plate sensor, even though we increased the number of plates we maintained the definitions of the swipe and hover gestures defined originally on the 4-plate sensor. To this end, we define multi-plate sub-regions according to their location with respect to the locations of the original plates in the 4-plate sensor. However, we are afforded the flexibility to dynamically reshape the sub-regions most appropriate for each particular differential measurement. For consistency in nomenclature in discussing the two sensors (4-plate and 12-plate), we define dynamic “virtual sensors”, 0 through 3, of the 12-plate sensor as the dynamic sub-regions described. For instance, in the 12-plate sensor a stage = $C_1 + C_2 - C_3 - C_4$ would imply that plates 1 and 2 are assigned to the positive terminal and plates 3 and plate 4 are assigned to be the negative terminal. For the above stage, $C_1 + C_2$ would comprise a virtual sensor and $C_3 + C_4$ would comprise a second virtual sensor. In our evaluation, we use two configurations of virtual sensors for the 12-plate sensor in Figure 6(a) and Figure 6(b). The configurations in
Fig. 6. (a) This figure demonstrates sensor configuration 1 for the 12-patch system. This configuration places the virtual sensors in the corners so that it can use more sensor plates per stage, creating a greater coverage area. Each of the six stages used for the signal processing is identified by the plates connected to the positive (green) and negative (red) terminals. (b) This figure demonstrates sensor configuration 2 for the 12-patch system. This configuration places the virtual sensors along the edges, thereby reducing the number of “reaching” gestures. This most closely approximates the processing used for the 4-patch system. Each of the six stages used for the signal processing is identified by the plates connected to the positive (green) and negative (red) terminals.

Figure 6(a) place the virtual sensors towards the corners of the sensor array and in Figure 6(b) the virtual sensors are placed towards the edges. We chose these two configurations because our conversations with subjects with limited mobility have revealed that some users can perform gestures by moving their hands to the edges of the sensor array while others are more comfortable performing gestures that span the corners of the sensor array. The sub-figures in Figure 6(a) and Figure 6(b) illustrate the stages. The + sign depicts the capacitor plates that are used as positive terminals while the - sign depicts the plates that are negative terminals.

VI. SYSTEM EVALUATION

The main goal of this system is to provide a low-power, accurate, and real-time method of controlling a smart-home through gesture recognition for assistive purposes, especially those with UE limited mobility. To demonstrate this, we have focused the evaluation of this system on the following questions:

(1) Does the system accurately classify gestures across all subjects? (2) What is the average energy consumption of the system, and how does that compare to the baseline system which does not utilize a hierarchical signal-processing architecture? (3) What trade-offs exist between recognition accuracy and the type and extent of training that must occur? In answering these questions, we present a set of micro-benchmarks which give a more detailed representation of the power-consumption and latency of the different subsystems within the prototype system.

A. Experimental Setup

The experimentation was performed using five adult able-bodied subjects and a single subject with a C6 spinal cord injury. The five able-bodied subjects act as a baseline for evaluating the accuracy of our gesture recognition system. Our evaluation on the limited mobility subject helps us understand the limitations of our system when applied to the target population group. We discuss our results from our limited mobility subject separately in Section VI-A. The experimental setup consisted of each subject wearing the CSA on their thigh, and performing a set of “swipe” and “hover” gestures. Each of the subjects performed an average of 180 gestures, which consisted of between 9-12 gesture sets, with each set containing each of the 16 gestures defined below. Using the four plates in the initial prototype illustrated in Figure 1, the “swipe” gestures are performed as the combinatorial movement from each plate to each other plate. More formally, all combinations of $i \rightarrow j$, where $i \neq j$, and $i$ and $j$ are the plates numbered from 0 through 3, representing a full 12 gestures. The gesture set included 4 “hover” gestures, which are denoted by the plate numbers $\{0, 1, 2, 3\}$. Note that for the twelve plate CSA, we used the configurations defined in Section V to design the four virtual sensors. Each subject was allowed time to get accustomed to the system with practice gestures before training began. All of the accuracy results for the able-bodied subjects comes from this single session. To accomplish this, the training and testing were performed off-line using cross-validation. Below, we present our micro-benchmarks from the prototype system, followed by accuracy results from the four plate prototype system, and system trade-offs. In our experiments with the limited mobility subject, we also compare the accuracy of recognizing the gesture when using the four plate and twelve plate configurations.

B. Micro-Benchmarks

Tables I and II present an evaluation of the energy consumption and latency of the different subsystems within the prototype system. The need for a hierarchical signal processing architecture is apparent based on the energy consumption. On average, the Bluetooth module consumes an order of magnitude more energy than the micro-controller which consumes another four times more than the capacitance measurement IC
Fig. 7. (a) This graph depicts the relative accuracy of three different classification methods: Nearest Neighbor Classifier, Decision Tree Classifier, and Naive Bayesian Classifier across our five subjects. Although the difference is slight, the 1NN algorithm maintained a higher overall accuracy than the other two. Error bars are included as the standard deviation of accuracy per algorithm. (b) A confusion matrix demonstrating the accuracy of the 1NN algorithm across all gestures and subjects. (c) This graph demonstrates the classification accuracy as a function of the size of the 1NN codebook. The accuracy increases asymptotically with the number of training sets. Error bars are included as the standard deviation of accuracy per user. (d) This plot demonstrates the accuracy of each subject for aggregate training (data from all subjects is considered) against personalized training (data only from that particular subject is considered). Personalization results in slightly increased accuracy and decreased variance on a per-user basis.

in its low-power measurement mode. This motivates the need to duty cycle the micro-controller and BLE module as often as possible, and only waking each piece of the hierarchy as needed. However, this design is useful only if the transition latency from low- to high-power tiers is reasonable for a real-time system. The latency reported by Table II demonstrates that the major component of delay in the system is in performing the machine learning algorithm, which takes on average little more than a quarter of a second on a 1 MHz micro-controller, demonstrating the efficiency of the implementation.

C. System Accuracy

This first accuracy evaluation focuses on the initial trial which compares the accuracy of classification of the five able-bodied subjects using the 4-plate prototype. Figure 7(a) demonstrates the relative classification accuracy for three machine learning algorithms (Nearest Neighbor Classifier, Decision Tree Classifier, and Naive Bayesian Classifier) across the five subjects. These three classifiers were chosen because they represent a wide computational range of needs. The Bayesian classifier and decision tree classifier require an amount of training that may not be feasible on a micro-controller. Each of these classifiers, however, are computationally efficient enough to operate on the micro-controller after their training phase. Nearest Neighbor can be completely implemented on the micro-controller so long as the number of gestures needed in the codebook are not substantially large enough to consume the memory. The accuracy of each of the three algorithms is around 90%, but a 1NN algorithm performs slightly better for our subjects with an average accuracy of approximately 93%. Therefore, for the real-time classification in our prototype system we implemented the 1-Nearest Neighbor classification.

D. Energy Consumption

Figure 8 demonstrates the energy consumption of two different systems; one which utilizes a hierarchical wake-up and data processing scheme, and a second (termed baseline) which processes all data on the micro-controller and uses the BLE only to send gestures. The baseline system consumes much more energy for the observation and measurement IC because the chip does not process the raw data, and cannot make decisions about whether to be in its low- or high-power state. Similarly, the baseline micro-controller is in an
always on state, consuming a near constant 500 uA, where the hierarchical system is able to duty cycle the micro-controller to reduce energy consumption by over 80% compared to the baseline.

The figure illustrates the average power consumption of the prototype at different gesturing frequencies, demonstrating that the energy consumption is partially dependent upon the frequency, but asymptotically converges to a baseline consumption near 500 uA. From these measurements, we present the following three conclusions. (1) The absolute power consumption of the system is very low for a gesture recognition system, and is close to 525 μA (1.7 mW) when gesture frequency is once every 2 minutes (which for the given application is a very high rate). Given a 1000 mAh battery, our prototype system could theoretically last for 2000 hours (83 days) on a single charge (though this would be diminished due to self-battery drain). (2) Our hierarchical signal processing allows for the system to consume approximately 4 times lower power than the baseline system, which utilizes already low-power components. (3) We find that the primary energy consumers in the system are the BLE module and the observation and measurement IC. These two energy consumptions can be addressed through the use of application specific hardware. The amount of information being transferred wirelessly is a single byte at most every 3-5 seconds. This would allow for the use of other wireless communications to reduce energy, including ambient back-scatter [37] communication, which could reduce the wireless communication energy consumption altogether. The measurement IC could also be implemented in low-power analog sub-threshold circuits to considerably reduce the energy consumption.

E. System Trade-Offs

This section sets out to investigate trade-offs in training size and accuracy, especially considering the effect of personalized vs aggregate gesture sets. Figure 7(c) demonstrates the first of these trade-offs, showing the accuracy for each user as the training size increases. Across each of the five subjects, the accuracy increases as the training size increases, but saturates after approximately five training sets. This is beneficial, as it limits the amount of training that a user must perform, and the amount of processing that must occur to perform the 1NN calculation. The second trade-off, aggregate vs. personalized training, is illustrated in Figure 7(d). The personalized training (i.e. where only training from that particular subject is used for classification) has a slightly higher average accuracy than the use of aggregate training (i.e. training data from all subjects is used). Additionally, the variance of accuracy in subjects 1 and 4 when using aggregate training data is considerably larger than the variance within the personalized set. These two subjects have proportions of their forearms and legs that are different from the other subjects, which could contribute to capturing different motion artifacts that were not captured in the other subjects. The use of personalized training allows for data to fit to the individual, resulting in better overall accuracy performance, which represents a fundamental design principle for our system.

1) User Study on a Subject With Impaired Mobility: To validate the performance of our system on subjects belonging to the target population group, we performed an evaluation on a single subject who has an upper extremity mobility impairment. This individual has a C6 Spinal Cord Injury which has resulted in total loss of mobility below the waist, and limited mobility in the fingers, wrists, and arms. Our initial discussion and evaluation with this individual identified that the hovers and swipes that comprise our gesture set are both within the individual’s ability and comfortable to perform over a long period of time. During the evaluation session, the individual was trained on the system by performing two sets of each gesture and receiving verbal instructive feedback. The individual then performed ten of each gesture on the 4-plate system for a total of 160 gestures. The subject then was given a short break before being asked to perform five of each gesture on each of the two 12-plate configurations (see Section V for description of the configurations) for a total of another 160 gestures (80 for each configuration). The collected data set was divided into 5 sets and was randomly divided into a training and testing set comprised of 3 and 2 sets respectively.

Figure 9(a) graphs the accuracy of Inviz in recognizing gestures for the 4-plate and 12-plate sensors. For the 12-plate sensor accuracy results are shown for both configurations described in §V. We can make two observations from the figure. First, the accuracy of gesture recognition improves as we use more plates. For instance, the accuracy of
recognizing the gestures on the 12-plate sensor is 80(+/−)3%. For the 4-plate sensor the accuracy of Inviz is lower than the healthy subjects (≈ 70%). We delve deeper into understanding the causes of the inaccuracies using the timestamped videos for the subject performing the gestures. We derived three major artifacts specific to this quadriplegic individual that caused the inaccuracies.

First, when considering extension motions such as moving the hand from the bottom plate to the top plate, the ideal movement maintains a greater separation between the forearm and the sensor plates compared to the hand. However, this subject’s forearm stays near the array as compared to the hand. This is due to his reduced ability to perform wrist extension. The phenomenon is demonstrated in Figure 10 (a). This type of movement biases capacitance readings in a near and far sensor pair towards the near sensor, potentially causing imbalanced thresholds that need to be met for the recognition algorithm. This is further illustrated in the confusion matrix for the subject in Figure 9 (b). From the matrix, we observe that the classification for this subject was diminished for gestures that extended over the top of other pads (e.g., hovering over plates 0 and 1 and moving from plate 1 to either 0 or 2). An example data stream which demonstrates this bias is shown in Figure 11. The figure illustrates two expected streams and the errors that are created by biased sensor readings from the presence of the forearm.

Secondly, the subject has small thighs, in part because of atrophy of the thigh muscles from disuse. This causes the sensor array to contour the leg in a more convex shape than the other healthy subjects. In order to facilitate correct left-to-right and right-to-left motions, the individual pronates and supinates his hand with muscle groups near the elbow to keep the hand near to the array. This motion is demonstrated in Figure 10 (b). Some individuals with upper extremity mobility impairments do not have the ability to perform this type of arm rotation based on upper arm rigidity and therefore have difficulty tracking the contour of the leg, and will have to mimic this motion by propagating shoulder movements, thereby creating more variation in the amplitude of peaks and potentially reducing the accuracy of moving above the correct plates.

Finally, the subject tended to face the side of his hand opposite to the thumb toward the sensor array while our healthy users all faced their palm towards the array. In discussion with this individual it was revealed that this was to facilitate moving through each gesture so that the opposite leg would not be a burden. However, a user who does not have the ability to retract fingers through tenodesis may choose to maintain this position so that their fingers do not drag along the array. This position has a thinner profile than a downward facing palm, creating a modest increase in peak separation and decrease in peak width comparatively, which can create error within the feature vectors and potentially causing the gesture recognition algorithm to misclassify the gesture.

The above artifacts points to two limitations of our prototype Inviz system. First, the sensor must be built in proportion to the user. A larger or smaller than required sensor array causes the gestures to be performed incorrectly. Second, the observations point to the need of a more robust statistical learning approach like Hidden Markov Models [31] for gesture recognition. We are pursuing the first avenue as future work, and have utilized a different mechanism to consider the second approach [35]. Furthermore, we plan to further investigate the impact of the distance of the hand from the sensors and the influence of disturbances in the environment on gesture recognition accuracy.

VII. CONCLUSION

In this paper we present an event-driven, low-power, wearable capacitive gesture recognition system as an assistive device for individuals with upper-extremity mobility impairments. The use of textile-based capacitive sensors integrated into items of everyday living allows the flexibility to use the system without issues of occlusion or obtrusion. Our system uses a replicable hierarchical sensor processing algorithm to break down computation into different power tiers and maximize responsiveness at a very low average energy consumption (<750 μA). We have implemented and evaluated a fully functional end-to-end prototype of the system in terms of latency, power consumption, and accuracy as an assistive home-automation cyber-physical system to demonstrate that the system is capable of accurately (>90%) classifying gestures in real-time (<1 s latency) for home use.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or Microsoft.
REFERENCES


Gurashish Singh (M’13) received the B.S. degree in computer engineering from the University of Maryland, Baltimore County (UMBC), Baltimore, MD, in 2014, where he is currently pursuing the M.S. degree in computer engineering, with research on sensors, biomedical systems, embedded systems, and very large-scale integration. He has taken an internship with Altera Corporation, San Jose, CA, exploring heterogeneous computing and was a Teaching Assistant at UMBC. He received the Runner-Up Best Demo Award at IEEE PerCom 2015 and an NSF GRFP Honorable Mention Award.

Alexander Nelson (S’13) received the B.S. degree and the M.S. degree in computer engineering from the University of Arkansas, Fayetteville, AR, in 2012 and 2013, respectively. He is currently pursuing the Ph.D. degree with the University of Maryland, Baltimore County, Baltimore, MD, with research on embedded real-time systems and pervasive computing. He is a member of IEEE-Eta Kappa Nu and Tau Beta Pi, and he received the Runner-Up Best Demo Award at IEEE PerCom 2015 and an NSF GRFP Honorable Mention Award.

Sheung Lu (S’15) is currently pursuing the B.S. degree in computer engineering from the University of Maryland, Baltimore County (UMBC), Baltimore, MD. He was a Teaching Assistant at UMBC and has an internship at the MIT Lincoln Laboratory and Texas State University–San Marcos. He joined the ECLIPS Research Cluster in 2014. His research interests include embedded systems, sensors, and very large-scale integration.
Ryan Robucci (S’00–M’10) received the B.S. degree in computer engineering from the University of Maryland, Baltimore County (UMBC), in 2002, and the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 2004 and 2009, respectively. He is currently an Assistant Professor with the Department of Computer Science and Electrical Engineering, UMBC. His current research interests include low-power, real-time signal processing with reconfigurable analog and mixed-mode very large-scale integration circuits; CMOS image sensors; sensor interfacing and networking; image processing; bioinspired systems; and computer-aided design and analysis of complex mixed-signal systems and hardware security. He received the ISCAS Best Sensors Paper Award in 2005 and the Runner-Up Best Demo Award at the IEEE PerCom 2015.

Chintan Patel (S’99–M’05) received the M.S. degree in computer science and the Ph.D. degree in computer engineering from the University of Maryland, Baltimore County (UMBC), in 2001 and 2004, respectively. He is currently an Assistant Professor with the Department of Computer Science and Electrical Engineering, UMBC. He specializes in very large-scale integration design and test. Specifically, he is involved in power supply modeling, noise estimation, current measurements circuits, and hardware security. He has received the Runner-Up Best Demo Award at the IEEE PerCom.

Nilanjan Banerjee received the B.S. degree in computer science and engineering from the Indian Institute of Technology (IIT) Kharagpur, and the M.S. and Ph.D. degrees from the University of Massachusetts, Amherst, in 2004. He is currently an Associate Professor of Computer Science and Electrical Engineering with the University of Maryland, Baltimore County, where he currently directs the Mobile, Pervasive, and Sensor Systems Laboratory. He received the Best Undergraduate Thesis Award from IIT Kharagpur in 2004. He also received the Yahoo! Outstanding Thesis Award from the University of Massachusetts, Amherst, in 2009. He is recipient of an NSF CAREER Award and a Microsoft Research Software Engineering Innovations Award.