

Robust, Low-cost, Non-intrusive Sensing and Recognition of Seated Postures

Bilge Mutlu¹ Andreas Krause² Jodi Forlizzi^{1,5} Carlos Guestrin^{2,3} Jessica Hodgins^{2,4}

(1) Human-Computer Interaction Institute, (2) Computer Science Department,

(3) Machine Learning Department, (4) Robotics Institute, (5) School of Design

Carnegie Mellon University

5000 Forbes Avenue Pittsburgh, PA 15213 USA

{bilge, krausea, forlizzi, guestrin, jkh}@cs.cmu.edu

ABSTRACT

In this paper, we present a methodology for recognizing seated postures using data from pressure sensors installed on a chair. Information about seated postures could be used to help avoid adverse effects of sitting for long periods of time or to predict seated activities for a human-computer interface. Our system design displays accurate near-real-time classification performance on data from subjects on which the posture recognition system was not trained by using a set of carefully designed, subject-invariant signal features. By using a near-optimal sensor placement strategy, we keep the number of required sensors low thereby reducing cost and computational complexity. We evaluated the performance of our technology using a series of empirical methods including (1) cross-validation (classification accuracy of 87% for ten postures using data from 31 sensors), and (2) a physical deployment of our system (78% classification accuracy using data from 19 sensors).

ACM Classification H.5.2 User Interfaces: *Haptic I/O, User-centered design*. J.7 Computers in other systems: *Consumer products*.

General Terms Algorithms, Design, Human Factors

Keywords: Haptic interfaces, sensor technologies, machine learning, personal technologies, ubiquitous computing.

INTRODUCTION

Many people spend a large part of their day sitting in an office chair, lounge chair or car seat. The ubiquity of seated activities in the workplace, home, and while commuting makes seating an interesting context for human-computer interfaces, specifically in ubiquitous computing applications and multi-modal user interfaces. While seating related issues, in particular discomfort and pressure sores, have received particular attention in military [9], workplace [21], assisted living [3, 28, 37] and mobility contexts [1, 23, 38], only a few have suggested seating as a part of a user interface [24, 29, 40, 41, 44].

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'07, October 7–10, 2007, Newport, Rhode Island, USA.

Copyright 2007 ACM 978-1-59593-679-2/07/0010 ...\$5.00.



Figure 1: On the left is the commercially-available Tekscan ConforMat system (2048 sensors) installed on a Herman Miller Aeron Chair. On the right is our deployed system (19 sensors) installed on the same model of chair.

In this paper, we consider the problem of recognizing *posture* which has been shown to be a primary measure of sitting comfort (and discomfort) [12] and a predictor for attention [24, 29], fatigue [18], and physical wellness [33]. We assess posture using a set of pressure sensors deployed in the seat and back of a chair. We have identified five domains where seated posture recognition could play a vital role.

Home In the home, a robotic chair could identify a pattern of postures and adjust itself to support the user's postural needs. For instance, if the user is falling asleep in a chair, the chair could identify the change in the user's posture over a window of time, infer that the user wants to sleep, and adjust the headrest appropriately [13].

Office Identifying posture patterns could help avoid problems caused by sitting in unhealthy postures for extended periods. Users could be made aware of these patterns and provided with recommendations.

Classroom Through identifying a pattern in a student's posture, an intelligent tutor could infer whether the student is interested or attending.

Automotive Posture information over a window of time could be used to infer the driver's level of drowsiness and attention. Such an application might improve the robustness of existing vision-based technologies that are sensitive to lighting conditions and changes in the driver's appearance.

Assistive Technology Posture information could be used by an intelligent wheelchair to help avoid seating ulcers. The chair could make minor adjustments on its surface or advise the user to change postures.

In this paper, we present an approach to seated posture recognition that provides a robust, inexpensive system with near-real-time prediction performance. Our methodology involves iterative design, where we (1) improve the design of our system through a series of experiments (using data produced by previous work [40, 41, 44]), and (2) prototype and evaluate our system design (with data collected from naive human subjects). Our final design displays accurate near-real-time classification performance on *unfamiliar* subjects by using a set of carefully designed, subject-invariant signal features.

More specifically, we provide the following contributions:

- **Robust generalizability:** A methodology for classification of postures, which achieves robust recognition on *unfamiliar* subjects.
- **Low cost:** A near-optimal sensor placement approach, which achieves classification rates similar to those reported in the literature [40, 41, 44] using less than 1% of the sensors (3% of the cost of a commercially available system).
- **Near-real-time performance:** A deployment of our classifier, which works in near-real-time (10Hz) on a standard desktop computer.

We evaluate the performance of our system using a set of empirical experiments including cross-validation using a large data set provided by the authors of [40], a physical implementation of our system to evaluate deployed performance, and cross-validation using data collected from 20 naive users on our deployed system.

The next section provides a summary of related work. The following section provides detail on the techniques we employed to develop our classification technology. We then describe the empirical evaluations we conducted to assess the performance of our technology. In the last two sections, we highlight some of the open questions related to our work and summarize our contributions.

RELATED WORK

Posture recognition has been used to detect irregular behavior, infer emotional state, and for input to and control of HCI applications. In this section, we first describe the related work by application domain: *Home*, *Office*, *Classroom*, *Automotive*, and *Assistive Technology*, and then discuss the most closely related work in more detail.

Home Sato created a *Robotic Room* that obtained posture information through pressure sensors embedded in a bed to detect and support the activities of a room’s inhabitants [35]. Panini and Cucchiara developed a vision-based posture detection system for home automation applications [31].

Office Pellegrini and Iocchi developed a posture classification algorithm for activities in an office context [32].

Classroom Kapoor and colleagues developed a learning companion that detected a student’s emotional state using seated posture information in combination with other measures to facilitate learning [24, 29]. D’Mello and colleagues integrated a seated posture-based affect recognition system into their intelligent tutoring system *AutoTutor* [14].

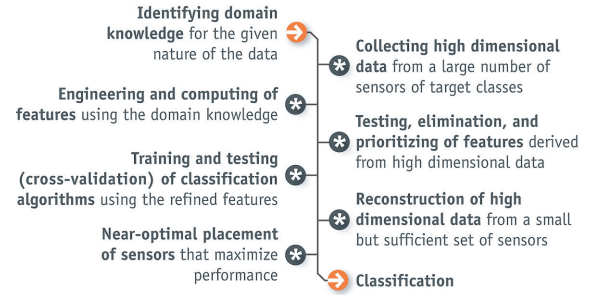


Figure 2: Our method for developing a robust, fast, low-cost learning system for recognizing classes in high-dimensional data.

Automotive Several papers report vision-based systems using seated posture information to detect driver fatigue [18] and to deploy *smart* airbag systems [8, 42].

Assistive Technology A system developed by Prado and colleagues used posture information to detect falls and follow physical activity patterns of elder users [33]. Aoki and colleagues used body position and posture to detect irregularities in human behavior [2]. Harada and colleagues used pressure sensors embedded in a bed to detect postures of bedridden users [22].

We detect seated postures using pressure sensors. Hence, we focus the rest of this section on similar technologies. The main body of work on detecting seated postures using a non-vision-based approach is by Tan and colleagues [40, 41, 44]. Their work classified a predetermined set of postures using a high-fidelity pressure sensor placed on an office chair. This research demonstrated significant potential for using posture recognition as an input modality with *familiar* subjects (subjects on which the posture recognition system was trained). However, the classification accuracy on *unfamiliar* subjects was either lower.

Sensor placement is crucial for high recognition rates and low hardware cost. The traditional approach to sensor placement assumes that sensors have a fixed sensing range. The best placement is then found by solving an *art gallery problem*¹ [19]. This purely geometric approach, however, does not take into account uncertainty and the specific correlation structure present in the posture recognition context. An alternative, statistical approach has been to learn a *Gaussian process* model which represents uncertainty about the area of sensing. This model is then used in order to optimize the positions of sensors [11]. Caselton and Zidek proposed *mutual information* to find sensor placements that are most informative about areas that are not directly observed by the placed sensors [6]. Recently, Krause and his colleagues introduced an approximation algorithm that finds a set of *near-optimal* sensor placements that are most informative about locations with no sensor information through maximizing an *information gain* criterion [26]. Our methodology adopts the approximation algorithm presented in this work.

METHODOLOGY

In this paper, we present a novel method to recognize discrete, static seated postures (Figure 2). The novelty in our approach is twofold. First, we use domain knowledge to engineer features that are highly predictive of the data. These domain-specific features outperform the use of generic di-

¹In the art-gallery problem, a minimum number of security cameras with fixed ranges has to be chosen in order to fully cover the gallery ground [7].

dimensionality reduction techniques such as Principal Component Analysis (PCA) in achieving linear separation. Second, our method performs a near-optimal sensor placement, which allows us to greatly reduce hardware cost and computational complexity. Using this technique, our system achieves superior performance to existing work using less than 1% of the sensors. Near-optimal placement also outperforms random and uniform placements in robustness and classification performance.

Our goal is to recognize (classify) seating postures. As input, we use sensor data acquired from a collection of pressure sensors placed on the seat and back of a chair, as displayed in Figure 1. At a given time, sensor data is acquired, processed, and a label for the posture (such as *sitting straight*, *leaning left*, and *slouching*) is predicted, which can then be used as input to an adaptive system such as those described in the scenarios in the Introduction. Our system discriminates among a set of ten postures (Figure 3), which were suggested in previous work [40, 41, 44]. This set of postures does not directly relate to our proposed application domains. Instead, it provides a benchmark to compare the performance of our algorithm to the performances of previous work.

Learning in our problem context requires (1) a set of *features* that explain the posture data, and (2) an *algorithm* for learning and prediction of postural information. While there is a large literature on how a classifier can be learned from training data, both the training algorithm and the set of features should be selected with a deep understanding of the recognition problem. Therefore, as a first step in choosing an algorithm and creating meaningful features for our classification, we did extensive analyses of postural data.

Understanding Our Data

In the posture classification problem, at every time step t our system is given the sensor values x_1, \dots, x_m (in vector form $\mathbf{x} \in \mathbb{R}^m$), where m is the number of sensors. Our goal is to output a class label $y \in \{1, \dots, C\}$, indicating the sitter’s posture. The sensor values \mathbf{x} are referred to as *pressure maps* – Figure 3 presents examples of pressure maps for different postures, obtained with the Tekscan high resolution pressure sensor used in earlier work [40, 41, 44]. In classification, we are given a training data set $\mathcal{D}_{train} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, and we want to use it to learn a

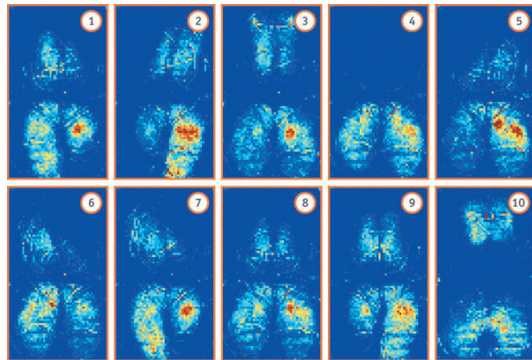


Figure 3: Sample data of pressure applied to the seat and back of an office chair. Samples display postures: (1) Left leg crossed, (2) Right leg crossed, leaning left, (3) Leaning back, (4) Leaning forward, (5) Leaning left, (6) Leaning right, (7) Left leg crossed, leaning right, (8) Seated upright, (9) Right leg crossed, (10) Slouching.

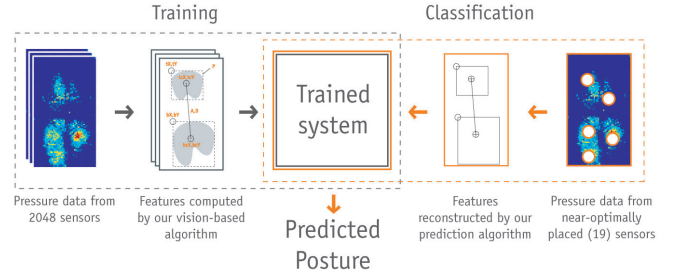


Figure 4: A diagrammatic illustration of our classifier.

function g such that $y \approx g(\mathbf{x})$, i.e., $g(\mathbf{x})$ is approximately the true class label, not just on the training set, but also on unseen input data. This requirement is known as *generalization*. In posture recognition, we specifically want the classifier to *generalize* between different subjects, i.e., to work well on *unfamiliar* subjects.

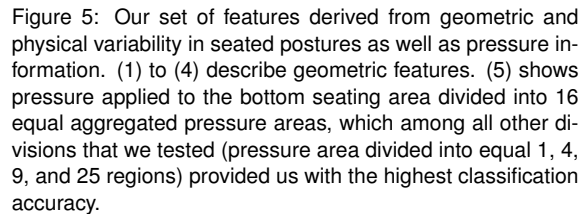
When we use the high-resolution Tekscan sensor, the number m of sensor values is very large, and we incur what is known as the *curse of dimensionality* [4] – generalization in a high dimensional data set is very difficult. Consequently, the two prior approaches in posture recognition [40, 44] deal with this problem by explicitly reducing the dimensionality of the input data using Principal Component Analysis (PCA) and Sliced Inverse Regression (SIR) respectively. These dimension reduction techniques find a low dimensional subspace which most effectively represents the training data, in the sense of minimizing the mean square reconstruction error.²

In addition to these prior approaches, we experimented with automatic feature extraction techniques such as maximizing *information gain* with respect to the class (posture) label. We also examined how effectively sophisticated classification algorithms could model the data without using prior feature selection. Poor generalization produced by these techniques led us to examine our data in more detail. We found that the variance across classes (i.e., postures) was much smaller than the variance across subjects, which produces high classification accuracies with *familiar* subjects but poor generalizability with *unfamiliar* subjects. To overcome this challenge, we explored how domain knowledge could be used to engineer a small set of features \mathbf{f} , and learn a classifier g mapping the feature values to the class label, i.e., $y \approx g(\mathbf{f})$. Our experimental results show that this approach improves generalization to unfamiliar subjects compared to existing work. Furthermore, we demonstrate that, contrary to all work to date, our approach does not depend on the Tekscan high-resolution sensor – we show how we can reconstruct the chosen features from a small set of near-optimally placed sensors, thereby achieving similar classification accuracy at a fraction of the material and computational cost. Figure 4 provides a diagrammatic illustration of our approach.

Features

In [44], a normalization step for preprocessing is proposed, in which the pressure maps are *cropped* to contain only areas where the sensor reading is non-zero. We argue that this cropping step discards valuable information about user’s posture. Furthermore, our experiments have shown that *where* pressure is applied is more informative about user’s posture

²While PCA is independent of the classification labels, SIR (as a generalization of Linear Discriminant Analysis) aims at selecting features (dimensions) that capture most of the class specific variance. Both, however, do not explicitly select features to maximize classification performance.



Our features represent two kinds of information; (1) where the pressure was applied at the bottom of the seat and at the back of the seat, and how these two areas are interrelated (illustrated with items (1) to (4) in Figure 5), and (2) how much pressure was applied in these areas (item (5) in Figure 5). Following this framework, we designed a set of 51 features and used a feature selection algorithm to find the most informative features based on classification accuracy [20] to identify a subset of 30 features that are the most informative of the class to be predicted.

1. The position and size of the bounding box of the pressure area on each surface (seat and back). In finding the bounding boxes, we first apply Hysteresis thresholding [5] to eliminate noisy edges for better detection of the pressure areas. The pressure areas are computed using a simple connected components algorithm [34] and finding the bounding boxes of the clustered areas.
2. The distance of the bounding boxes of the pressure areas at the bottom and the back to the edges of the seat.
3. The distance and angle between the centers of the pressure areas at the back and the bottom surfaces.
4. The centers, radii, and orientations of two ellipses fit to the pressure areas created by the sitter's buttocks. These ellipses are calculated using a Least-Squares ellipse fitting algorithm described in [15].
5. Pressure applied to the bottom area divided into 16 aggregated sub-regions. We decided on dividing the pressure area into 16 aggregated sub-areas empirically. This division outperformed other candidate divisions (into 1, 4, 9, 16, or 25 regions) in a cross-validation experiment using a Logistic Regression classifier.

We tested the performance of our features against features used by existing work. For instance, the method described in [40] uses Principal Component Analysis (PCA) to reduce the dimensionality of the data and extract the features (Principal Components) that are most informative of the data. A cross-validation experiment using our features outperformed the accuracy provided in [40]. We also tested linear separation performance achieved by our features \mathcal{F} against that of PCA. Figure 6 displays the results of this experiment where our features achieved a much lower number of misclassified instances.

After we extract the feature values from the training data, we train a classifier based on Logistic Regression (LR), a classification technique which finds a linear decision boundary to separate the classes. For instance, to separate two postures, the classifier finds a vector of coefficients $\mathbf{w} = (w_1, \dots, w_m)$ such that $\mathbf{w}^T \mathbf{f} = \sum_{i=1}^m w_i f_i$ is positive for feature values $\mathbf{f} = (f_1, \dots, f_m)$ from one posture, and negative for the other. For k postures, LR models the conditional likelihood of posture j , $P(y = j \mid \mathcal{F} = \mathbf{f})$ given features $\mathcal{F} = \mathbf{f}$ as

$$P(y = j \mid \mathcal{F} = \mathbf{f}) = \frac{\exp(-\mathbf{w}_j^T \mathbf{f})}{\sum_{l=1}^k \exp(-\mathbf{w}_l^T \mathbf{f})}.$$

One way to find the coefficients \mathbf{w}_j is maximum conditional likelihood estimation, i.e., to find the coefficients which maximize the likelihood of observing all labels $y^{(i)}$ for all features $\mathbf{f}^{(i)}$, where i is the index of the training example. Another approach, which often is more robust, is to assume a *sparse* representation of \mathbf{w}_j . In this sparse representation, each weight vector \mathbf{w}_j (for the j -th posture) depends only on a small set of features, i.e., has only few non-zero entries. Note that these non-zero entries can be different for each posture, i.e., every posture can be characterized by a different subset of features. This sparse approach favors *simpler* models, which leads to improved generalization to unseen data. The *SimpleLogistic* algorithm [27] finds such a sparse representation by using the LogitBoost algorithm [17] to optimize an additive logistic regression model; each ad-

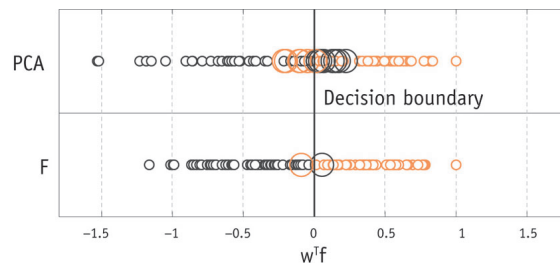


Figure 6: Best linear separation achieved using PCA (top) and our features (F at the bottom), for distinguishing postures (2) Right leg crossed, leaning left and (9) Right leg crossed. Big circles mark misclassified examples. For both PCA and F, we train a Logistic Regression classifier, and project the examples (i.e., compute $\mathbf{w}^T \mathbf{f}$) along the normal vector \mathbf{w} of the respective decision boundary. F achieves better separation.

ditive component is a simple linear function h depending on a single feature $h(\mathbf{f}) = w_l f_l$ for some scalar weight w_l and index l . The additive components are optimized on an appropriately weighted training set. The weights for each new component are chosen in order to minimize the residual classification error, i.e., a new weight is added to a new feature, such that the number of examples (misclassified using the existing weights) is minimized. This iterative procedure stops when the classification accuracy cannot substantially be improved by adding more features. Details of this approach can be found in [17, 27]. We selected this approach after empirical comparison with several other classification techniques.

In addition to robust, high classification accuracy, the advantage of using Logistic Regression is that instead of simply outputting a class label, it outputs a probability distribution over class labels, which contains an estimate of the predictive uncertainty. This notion of confidence may be important for the application using the posture information.

Near-Optimal Sensor Placement

Although the approach described in the previous section achieves high classification accuracy, it requires input data from a high resolution pressure sensor, in order to accurately compute the features discussed above. The cost of this sensor provides a significant burden for its large scale deployment, e.g., in automobiles, wheelchairs, or office chairs. In addition, the features need to be computed from the high dimensional data in near-real-time, demanding high computational performance. In order to avoid these problems, our methodology seeks to replace this high resolution sensor with a small set of pressure sensors, strategically placed in order to provide similar classification accuracy. If \mathcal{V} is the set of all possible sensor locations (e.g., the grid cells of the Tekscan sensor), we want to find a subset $\mathcal{A} \subseteq \mathcal{V}$ of these sensors, such that, given the pressure $\mathbf{x}_{\mathcal{A}}$ at only these locations, our system can classify the posture as accurately as possible.

The high performance of our features motivated the following approach towards selecting near-optimal sensor locations: Our algorithm selects sensor locations, which allow us to *reconstruct* the features \mathbf{f} (e.g., the bounding box of pressure, angle between mass centroids, etc.) as accurately as possible. After reconstruction, we use the classifier trained on the features \mathbf{f} to classify the posture. In order to achieve the reconstruction, our classifier learns a probabilistic model that describes the correlation between the measured sensor values and the predicted feature values. This probabilistic model allows us to select sensor locations that minimize the *uncertainty* in the prediction of the feature variables.

More formally, we model a set of random variables $\mathcal{X}_{\mathcal{V}}$, where each $\mathcal{X}_s \in \mathcal{X}_{\mathcal{V}}$ for $s \in \mathcal{V}$ represents the sensor value obtained from the Tekscan sensor at location s . Furthermore, we define random variables \mathcal{F} that model the values of the features (that depend on the sensor values). If we assume that we have a statistical model for the joint distribution, $P(\mathcal{X}_{\mathcal{V}}, \mathcal{F})$ (we will describe how to fit such a model later), and if we know the values of all the sensors, $\mathcal{X}_{\mathcal{V}} = \mathbf{x}_{\mathcal{V}}$, we can compute the feature values $\mathcal{F} = \mathbf{f}$ exactly, without any uncertainty. If we can only observe a subset $\mathcal{A} \subseteq \mathcal{V}$ of the sensor values, i.e., $\mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}}$, then the conditional probability $P(\mathcal{F} = \mathbf{f} \mid \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}})$ can be computed, and feature

values \mathbf{f} that maximize this conditional probability can be predicted. Even more importantly, this conditional probability distribution describes the *uncertainty* in the prediction, which we can quantify using the (conditional) Shannon entropy (c.f., [10]):

$$H(\mathcal{F} \mid \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}}) = - \int P(\mathbf{f} \mid \mathbf{x}_{\mathcal{A}}) \log_2 P(\mathbf{f} \mid \mathbf{x}_{\mathcal{A}}) d\mathbf{f}. \quad (1)$$

We want to select a subset of sensor locations, \mathcal{A} , such that this uncertainty is minimized, in expectation over all possible observations $\mathbf{x}_{\mathcal{A}}$:

$$\mathcal{A}^* = \arg \min_{\mathcal{A}} \int P(\mathbf{x}_{\mathcal{A}}) H(\mathcal{F} \mid \mathbf{x}_{\mathcal{A}}) d\mathbf{x}_{\mathcal{A}}. \quad (2)$$

This objective is equivalent to finding the subset of variables \mathcal{A} which has the highest *information gain*

$$\text{IG}(\mathcal{A}; \mathcal{F}) = H(\mathcal{F}) - H(\mathcal{F} \mid \mathcal{X}_{\mathcal{A}})$$

with respect to the feature variables (c.f., [10, 16]). In our optimization, we constrain the set \mathcal{A} to have a fixed number k of elements. Finding the optimal k -element subset maximizing the information gain is an NP-hard problem, but there is strong theoretical evidence [26] that in many practical applications the following greedy algorithm achieves near-optimal results. The algorithm starts with the empty set $\mathcal{A} = \emptyset$, and iteratively adds the sensor location s^* which increases the information gain the most, i.e., adds

$$s^* = \arg \max_s \text{IG}(\mathcal{A} \cup \{s\}; \mathcal{F})$$

to the set \mathcal{A} . The procedure stops after k sensors have been added. It follows that the greedy algorithm, given a set of sensors \mathcal{A} already selected, should pick the sensor s which maximizes $H(\mathcal{X}_s \mid \mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_s \mid \mathcal{F}, \mathcal{X}_{\mathcal{A}})$. This rule is quite intuitive – the algorithm prefers to add sensors which are as different as possible from the sensors already selected ($H(\mathcal{X}_s \mid \mathcal{X}_{\mathcal{A}})$ is large), but which are also relevant for predicting \mathcal{F} (i.e., $H(\mathcal{X}_s \mid \mathcal{F}, \mathcal{X}_{\mathcal{A}})$ is small, e.g., if \mathcal{X}_s is strongly correlated with \mathcal{F}).

Krause and Guestrin [26] proved that, under certain assumptions consistent with a wide range of real-world applications, this greedy procedure finds a set which is guaranteed to achieve at least a constant fraction $1 - 1/e$ (i.e., approximately 63%) of the optimal information gain. This guarantee is a worst-case bound; in practice, the solutions obtained are very close to optimal.

In order to use this algorithm for sensor placement, we need to specify a joint distribution $P(\mathcal{F}, \mathcal{X}_{\mathcal{V}})$. In our approach, we use the multivariate normal distribution; this distribution is fully parameterized by a mean vector μ and a covariance matrix Σ , i.e.,

$$P(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp \left(-\frac{1}{2} (\mathbf{z} - \mu)^T \Sigma^{-1} (\mathbf{z} - \mu) \right),$$

where $\mathbf{z} = (\mathbf{f}, \mathbf{x}_{\mathcal{V}})$, and $N = |\mathcal{V}| + |\mathcal{F}|$ is the total number of variables. These parameters can be estimated from the training data using maximum likelihood estimation. For a subset $\mathcal{A} \subseteq \mathcal{V}$, the conditional distribution of interest $P(\mathcal{F} \mid \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}})$ is found again to be multivariate normal, with mean $\mu_{\mathcal{F} \mid \mathcal{X}_{\mathcal{A}}}$ and covariance $\Sigma_{\mathcal{F} \mid \mathcal{X}_{\mathcal{A}}}$ as given by

$$\mu_{\mathcal{F} \mid \mathcal{A}} = \mu_{\mathcal{F}} + \Sigma_{\mathcal{F}\mathcal{A}} \Sigma_{\mathcal{A}\mathcal{A}}^{-1} (\mathbf{x}_{\mathcal{A}} - \mu_{\mathcal{A}}), \quad (3)$$

$$\Sigma_{\mathcal{F} \mid \mathcal{A}} = \Sigma_{\mathcal{F}\mathcal{F}} - \Sigma_{\mathcal{F}\mathcal{A}} \Sigma_{\mathcal{A}\mathcal{A}}^{-1} \Sigma_{\mathcal{A}\mathcal{F}}, \quad (4)$$

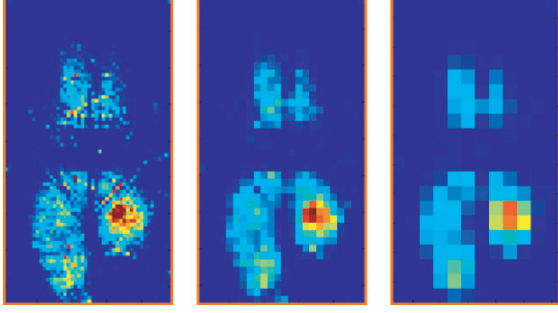


Figure 7: A posture instance in multiple resolutions. The leftmost image is the raw pressure readings. In the center is data points aggregated into 2×2 regions. The image in the right shows data points combined into 4×4 regions.

where for sets of variables \mathcal{B} and \mathcal{C} , the notation $\mu_{\mathcal{B}}$ refers to the entries of μ corresponding to the variables in \mathcal{B} . Similarly, $\Sigma_{\mathcal{BC}}$ is the submatrix of Σ which we get by selecting the rows corresponding to the variables in \mathcal{B} and the columns corresponding to the variables in \mathcal{C} .

Using this model, we can compute the conditional entropies

$$H(\mathcal{F}) = \frac{1}{2} \log_2(2\pi e)^N |\Sigma_{\mathcal{F}}|, \quad (5)$$

$$H(\mathcal{F} | \mathcal{X}_{\mathcal{A}}) = \frac{1}{2} \log_2(2\pi e)^N |\Sigma_{\mathcal{F}|\mathcal{A}}|. \quad (6)$$

Equations (1) and (2) can be efficiently computed in closed form.

Multi-resolution Sensor Placement

An alternative approach to selecting all pressure sensors from the same granularity is employing different types of sensors. We might, for instance, want small point sensors for detecting the boundary of the sitting region, but wider, flat sensors reporting the average pressure in a certain region. In order to allow this extension, we augment the set of variables $\mathcal{X}_{\mathcal{V}}$. Now, \mathcal{V} is the union of two (or more) parts, $\mathcal{V}_1, \dots, \mathcal{V}_r$, where each subset corresponds to sensors of a different resolution. Given the sensor values at the finest resolution (using, e.g., the Tekscan sensor), we can downsample the training data (i.e. Figure 7) to get coarser resolutions, i.e., by combining regions of 2×2 , 4×4 , etc. sensors, and averaging their values. Using this procedure, we can augment the training data set to simulate the values of coarser sensors. Our algorithm then learns a joint distribution $P(\mathcal{F}, \mathcal{X}_{\mathcal{V}})$ on this new augmented set of variables, and the sensor placement algorithm as described in the previous section will automatically determine which resolution to use.

Furthermore, we can also allow for different costs c_i for the different sensor types \mathcal{V}_i . Using a slightly more complicated algorithm, which combines the greedy algorithm (which now optimizes the benefit cost ratio) and a partial enumeration scheme, we can select near-optimal placements \mathcal{A} which maximize the information gain $IG(\mathcal{F}; \mathcal{X}_{\mathcal{A}})$ subject to a budget, a constraint on the total cost of the placement \mathcal{A} . Details on this procedure can be found in [39, 25].

EXPERIMENTS

We conducted two sets of experiments, one with data collected with the Tekscan sensor and one with data collected using our prototype sensor system.

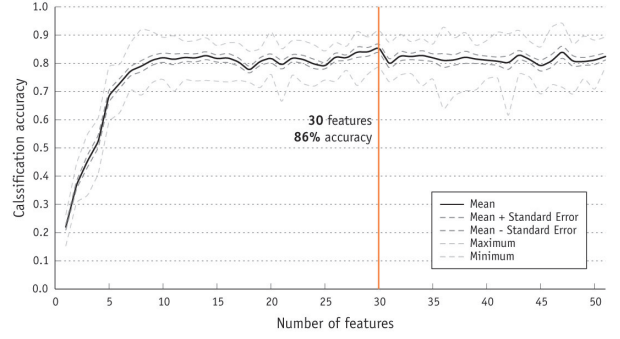


Figure 8: Ten-fold cross-validation results for change in classification accuracy as more features are added. The sequence of adding these features followed a ranking created by an SVM-based feature selection algorithm [20]. Vertical line indicates the cut-off point we chose.

Evaluation Using Tekscan Sensor Data

We performed our first set of experiments on the data set made available to us by the authors of [44] and [40], thereby making our results comparable to these existing approaches. The data set contains pressure data for ten postures, collected from 26 male and 26 female participants. The postures included are (1) *Left leg crossed*, (2) *Right leg crossed, leaning left*, (3) *Leaning back*, (4) *Leaning forward*, (5) *Leaning left*, (6) *Leaning right*, (7) *Left leg crossed, leaning right*, (8) *Seated upright*, (9) *Right leg crossed*, (10) *Slouching*. From each participant and posture, five examples were selected. Data was collected using two sheets of the Tekscan sensor, each providing 42×48 sensing elements, at a 10mm distance. The elements provide a 8-bit resolution for sensing pressure.

We used ten-fold cross-validation in our experiments that repeatedly selected a gender-balanced random subset of 23 male and 23 female participants for training, and evaluated our classifier on the remaining six participants (three male, three female). The reported results are average scores with respect to ten such random splits. This split enables us to assess the generalization capabilities *between* participants, which we refer to as the *multi-user* setting, using the terminology from [40]. The best performance reported by previous work in this setting was 79% [40].

We implemented the algorithms for computing the feature values in MATLAB, and used the *SimpleLogistic* classifier implementation from the Weka library [43]. We also implemented the greedy algorithm for sensor placement in MATLAB. To use this algorithm, we trained a multivariate normal model on the sensor values and the computed features on the training data using maximum likelihood estimation.

Feature Selection Our first experiment aimed at prioritizing the features on our feature set using rankings created by an SVM-based feature selection algorithm [20] and identifying a subset of these features that maximizes classification performance. To identify this subset, we conducted a ten-fold cross-validation experiment, which provided us with information on how the classification accuracy changed as features were incrementally included in the learning model, without overfitting to a particular test set. Figure 8 presents results on this feature selection experiment. We can see that the classification accuracy quickly increases. As the number of features reaches ten, the classification performance levels

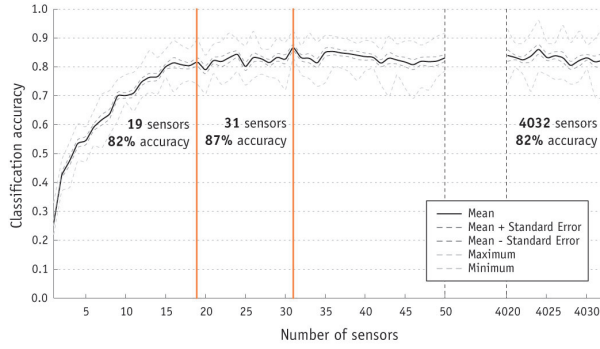


Figure 9: Ten-fold cross-validation results for the change in accuracy as the number of sensors increase. Best results were obtained with 31 sensors (87% accuracy), while we deployed 19 sensors, which produce the same accuracy (82%) as 4032 sensors do.

off and a maximum is reached at 30 features. Based on this analysis, we decided to use the first 30 features to train our classifier. Using this number, our classifier achieved an average (cross-validated) classification accuracy of 86%, compared to the trivial baseline of 10% if we guess the posture at random.

The top ten selected features, achieving an accuracy of 82%, included pressure readings from five aggregated pressure areas from the bottom of the seat, three of the variables that define the location of the pressure area at the back of the seat, and the angle and distance between the centers of the top and bottom pressure areas.

Sensor Placement After feature selection, we conducted experiments on selecting a sufficient number of sensors and the near-optimal placement of these sensors to achieve the targeted classification performance. We conducted a second ten-fold cross-validation experiment that looked at how classification accuracy changed as data from each sensor in our dataset were incrementally included in the learner. Our algorithm selected sensors at multiple resolutions of discretization, as described in the previous section. Figure 9 presents how the (ten-fold cross-validated) classification accuracy increases as more and more sensors are selected. However, the performance levels off quickly – even when placing 19 sensors, our classifier achieves an accuracy of 82% (equal to the accuracy obtained using all 4032 sensors). The best classification accuracy we obtained was **87%** using 31 sensors, which to our knowledge is higher than existing work using the same dataset, while using less than 1% of the sensors used in these work.

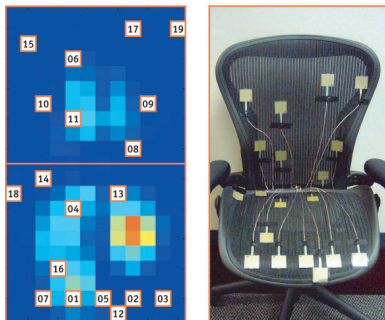


Figure 10: Final near-optimal placement of the sensors overlaid on an illustration of the downsampled data. The printed numbers on the sensor locations indicate ordering of sensors from the best predictor to the least. On the right shows our deployment of this near-optimal placement.

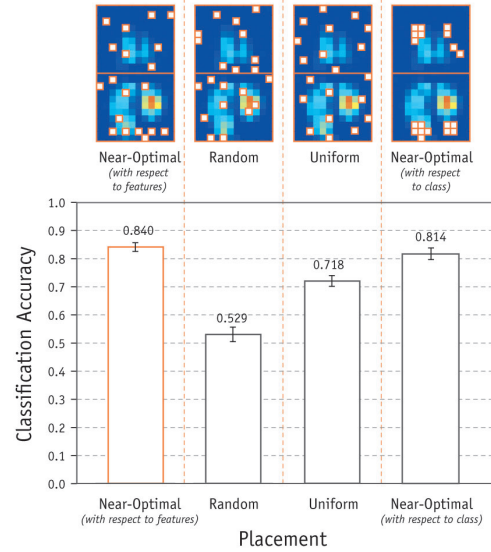


Figure 11: Cross-validation results that compare uniform, random, and near-optimal (with respect to class and features) placements of sensors on the chair surface. Images on the top row illustrate actual placements computed by these methods. The last placement selects sensors directly (without using the features) to maximize information gain.

We limited our final system to use 19 sensors, to be near-optimally placed on the bottom and back of the chair. Interestingly, all of the 19 sensors were chosen at the same level of discretization (4x4), aggregating pressure values from 16 sensors. Based on this result, we limited our hardware design to only sensors of this granularity. Figure 10 shows the near-optimal placement of these 19 sensors overlaid on sample aggregated data as well as on our final deployed system.

We also compared the effectiveness of our near-optimal placement with other possible placements. Therefore, another ten-fold cross-validation experiment compared classification performance of a *near-optimally* placed subset of sensors against *random* and *uniform* placements of the same number of sensors. Additionally, we compared the performance of our algorithm with a potentially more direct sensor placement approach which directly optimizes the information gain with respect to the class variable (instead of the feature values, as our approach does). Figure 11 displays the results of this experiment. Our near-optimal placement outperforms the uniform and random placements in classification accuracy³. As seen in Figure 11, it also performs better than the alternative, more direct placement strategy that does not use our features, again underlining the expressiveness of our chosen features.

Posture-Specific Performance Most misclassified labels were (3), *Leaning back*, with true positive and negative rates of 75% and 72% respectively, and (8), *Seated upright*, with rates 76% and 75%. This result is not surprising, since these postures are similar to several other postures in the data set, and there is high variance across subjects. Figure 12 shows the confusion matrix of the postures. We can see that postures (3), *Leaning back*, and (10), *Slouching*, were the ones which were most frequently confused, which is also intuitive due to the similarity of these postures.

³As the reader might have noticed, the cross-validation accuracies reported in different experiments are slightly different. This is due to different random splits done for cross-validation in different experiments.

class	1	2	3	4	5	6	7	8	9	10
1	235	0	1	0	0	0	18	3	2	1
2	0	240	0	0	5	0	0	0	15	0
3	0	0	187	5	1	2	0	28	0	37
4	0	0	3	244	1	0	0	10	0	2
5	1	5	2	7	241	0	0	4	0	0
6	0	0	0	16	0	222	5	14	3	0
7	26	0	0	0	0	3	231	0	0	0
8	2	0	24	28	5	3	0	196	1	1
9	0	5	0	2	1	2	0	0	250	0
10	0	0	32	3	1	0	0	3	0	221

Figure 12: Confusion matrix. Rows indicate true classes, columns refer to assigned labels. Class labels represent (1) Left leg crossed, (2) Right leg crossed, leaning left, (3) Leaning back, (4) Leaning forward, (5) Leaning left, (6) Leaning right, (7) Left leg crossed, leaning right, (8) Seated upright, (9) Right leg crossed, (10) Slouching.

We also looked at which features were selected (i.e., given non-zero coefficients in the posture specific weight vectors w_l) for the individual postures by the SimpleLogistic classifier. These weight vectors are very sparse – on average, every posture depends only on approximately three out of the 30 features. Inspection of these weights also gives valuable insight: For example, the two postures, (2) and (5), that included *Leaning left* are positively correlated with the horizontal bounding box location at the back, whereas the postures (6) and (7) that involved *Leaning right* are negatively correlated with this feature.

Reconstruction of $\mathcal{X}_{\mathcal{V}}$ We can also use the joint distribution $P(\mathcal{F}, \mathcal{X}_{\mathcal{V}})$ to reconstruct the unobserved sensor locations by estimating $P(\mathcal{X}_{\mathcal{V} \setminus \mathcal{A}} | \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}})$. Hereby, $\mathcal{X}_{\mathcal{V} \setminus \mathcal{A}}$ are the values of all the locations $\mathcal{V} \setminus \mathcal{A}$ where no sensors were placed, and $\mathbf{x}_{\mathcal{A}}$ are the measurements made by our sensors at locations \mathcal{A} .

Figure 13 presents examples of these reconstructed pressure maps. Images in the top row display the original, high resolution pressure maps acquired from the Tekscan sensor, while the ones in the bottom row show the reconstructed data using only the measurements of 19 near-optimally chosen sensors. We can tell through visual inspection that the reconstructed pressure maps are reasonably accurate, which indicates that the selected sensor locations capture the most important characteristics of the different postures.

We also performed an experiment, where we reconstructed the pressure maps as described above, then *deterministically*

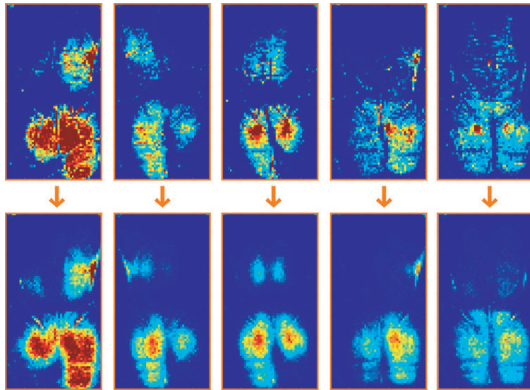


Figure 13: Reconstructed posture data examples on six postures. Top of each pair: original sensor values (4032 sensors); right image: reconstruction using only selected 19 sensors.

computed the features on the full resolution maps, and used the computed features for classification. Unfortunately, this approach leads to far worse classification accuracy of 54%, as opposed to 84% when probabilistically reconstructing the features directly.

Learning Algorithm Comparison A final evaluation of our method aimed at comparing different learning algorithms. Therefore, we trained *SimpleLogistic*, *Naive Bayes*, *Artificial Neural Network* (Multi-Layer Perceptron), and *Support Vector Machine* classifiers using our final feature set calculated by data from our near-optimally placed 19 sensors. Figure 14 presents the results of this experiment. *SimpleLogistic* produced higher classification accuracies than all other algorithms in this experiment.

Evaluation of Our Prototype Sensor System

Based on the results described in the previous section, we deployed a proof-of-concept prototype sensor system, as illustrated in Figure 1, and installed it on an office chair. We chose the same chair (the Aeron chair produced by Herman Miller) as used in the original data collection by the authors of [44] and [40]. The final system included 19 one-and-a-half-inch-square sensors near-optimally placed on the seat and back of the chair. We used FSR (Force Sensing Resistors) sensors produced by Interlink Electronics. Sensors were connected to a data acquisition board that read sensor values with the desired frequency and sent to a desktop computer via a USB connection. We implemented a Java application for near-real-time data acquisition, processing, and classification.

We used our deployed sensor system to evaluate the performance of our classifier in a real-world evaluation setting. We hired 20 naive subjects (ten male, ten female, college students aged between 19 and 34) to sit in the ten postures we used for training our classifier in ten trials (100 postures per participant). The order that the postures appeared in each trial was randomized to avoid learning or discomfort effects.

Classification Accuracy We analyzed the data collected from this deployment in several ways. In order to use the classifier trained on the high resolution data set, we need to calibrate the deployed sensors. We calibrated the sensors by performing linear regression, finding slope and bias for each deployed sensor to minimize the error in predicting the cor-

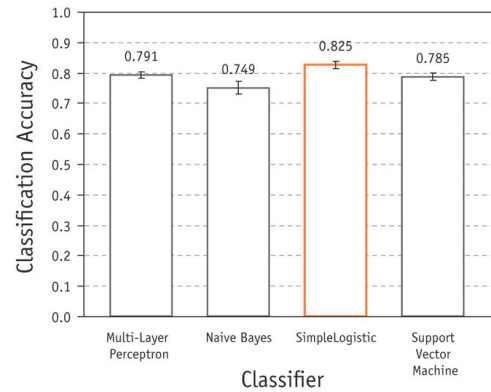


Figure 14: Ten-fold cross-validation results for classification accuracy with 4 classifiers (SimpleLogistic, Naive Bayes, Support Vector Machine, and Multi-Layer Perceptron) as well as Information Gain with respect to the target class variable.

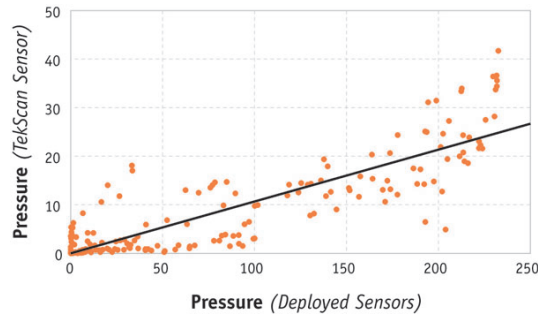


Figure 15: Regression analysis, comparing the of mean pressure values for each posture between the deployed sensors and the corresponding values of the Tekscan sensor.

responding values from the high resolution sensor. Because we did not have data from the same subjects on both the deployed sensors and the Tekscan sensor, we performed this regression analysis on the posture specific means, as presented in Figure 15. Due to the low-fidelity nature of the deployed sensors, their different signal response compared to the Tekscan sensor, and the variance in the subjects' postures, this calibration was very difficult. In fact, when we used the calibrated data from the sensors, and classified it using the classifier trained on the high resolution data, our accuracy was only 63%.

Not satisfied with this result, we *re-trained* the classifier on the data from the deployed sensor only. We conducted a ten-fold, gender-balanced, cross-validation experiment. In each cross-validation split, we trained a *SimpleLogistic* classifier on randomly selected nine male and nine female participants and tested on the remaining two participants. We used the 19 measured values directly as features. Our classifier achieved a (ten-fold cross-validated) classification accuracy of **78%**. Considering the fact that the chosen sensors have quite different and lower-fidelity responses than the high-cost Tekscan sensor, as well as the fact that we trained the classifier on only 18 (instead of 46) subjects, this classification accuracy is very promising, and shows that the proposed sensor placement technique can lead to high quality sensor placements at a small fraction of the cost of existing approaches.

Real-time Performance The second experiment was an exploration of the real-time performance of our deployed system. Real-time performance is particularly important in applications where posture information is required repeatedly to identify a temporal pattern in user's behavior (e.g. in detecting that someone is falling asleep, in measuring user's attention level). Our deployed posture recognition system can predict postures at 10 Hz on a fairly standard desktop computer, which we believe is a promising performance considering that no near-real-time systems on posture recognition is reported in the literature. We are also convinced, that if the classifier is implemented on a dedicated embedded device, far lower response times can be achieved.

Cost Analysis Through deploying our sensor system, we were also able to get an estimate of the cost of the system. The cost of only the sensors in our system is 100 USD compared to the Tekscan sensors commercially available for 3,000 USD. We argue that our low-cost sensor system will allow us to deploy the technology in different settings. Collecting data from a larger number of people and in a variety

of seating contexts will provide us with a deeper and more precise understanding of posture data and seated activities. In the next section, we discuss possible applications.

DISCUSSION

Our results demonstrate that even subtle posture differences can be reliably and robustly detected with low cost hardware. However, they are only a step towards the larger goal of extracting useful high level information, such as the users activity or comfort level. In this work, we limited our analysis to postures previously considered in the literature [40, 41, 44] for sake of comparison. As a next step, we plan to observe people in natural settings to identify the kinds of postures that people sit in during particular activities and quantify them in order identify and prioritize most important postures that need to be distinguished.

Our work to date focused only on classifying static postures. In order to extract activity or comfort information, we expect the temporal aspect of the problem to become very important. For example, even though people change postures rather frequently, such changes do not necessarily indicate a change in their activity. By looking at temporal sequences of posture changes, we might gain more information. On the other hand, a sitter's posture might change only slightly while the sitter shifts from reading a book to watching TV. Hence, in this case, additional input modalities would be needed to accurately identify the person's activity. Understanding the potential and limitations of when postural and temporal information are indicative of a user's activity is an important open question.

Another open question is to what extent training data obtained using a particular chair could be used for another chair. We believe that training of the system could be transferred across chairs within the same application domain (e.g. car seating) using a simple mapping, while new training would be required for applications where the properties of the seat changed significantly. However, further exploration is required to understand the limitations of such transformations.

CONCLUSIONS

In this paper, we presented a highly practical approach towards sensing and recognizing seated postures. Using a near-optimal sensor placement strategy, our approach achieves a classification accuracy of 87% in classifying ten postures on untrained subjects, a higher performance than that of previous work. Our system uses less than 1% of the deployed sensors compared to previous work, therefore drastically reducing hardware and computational cost. We used our methodology to build a prototype near-real-time sensing and recognition system using only 19 sensors. In a user study with 20 subjects, our low-cost prototype achieved 78% accuracy in the same classification task.

ACKNOWLEDGMENTS

This work was supported by NSF grants IIS-0121426, DGE-0333420, CNS-0509383, a gift from Intel Corporation and a Ford Motor Company Graduate Student Research Grant. Carlos Guestrin was supported by an IBM Faculty Fellowship, and an Alfred P. Sloan Research Fellowship. We would like to thank Professors Hong Tan of Purdue University and Lynne Slivovsky of California Polytechnic State University for providing their data set for comparison.

REFERENCES

- Agency for Health Care Policy & Research (AHCPR). Pressure ulcers in adults: Prediction and prevention, clinical practice guideline. *AHCPR Publication no. 92-0047*, 1992.
- Aoki, S., Iwai, Y., Onishi, M., Kojima, A., and Fukunaga, K. Learning and recognizing behavioral patterns using position and posture of human body and its application to detection of irregular states. *Systems & Computers in Japan*, 36, 13 (2005), 45–56.
- Bardsley, G.I. The dundee seating programme. *Physiotherapy*, 70, 2 (1984), 59–63.
- Bellman, R. *Dynamic Programming*. Princeton U. Press, 1957.
- Canny, J.F. A computational approach to edge detection. *IEEE Trans. on Pattern Anal. & Machine Intel.*, 8, 6 (1986), 679–698.
- Caseltan, W.F. and Zidek, J.V. Optimal monitoring network designs. *Statistics & Probability Letters*, 2, 4 (1984), 223–227.
- Chvatal, V. A combinatorial theorem in plane geometry. *J. of Combinatorial Theory, Series B*, 18 (1975), 39–41.
- Cheng, S.Y. and Trivedi, M.M. Human posture estimation using voxel data for “smart” airbag systems: issues and framework. *IEEE Intelligent Vehicles Symposium*, 2004, pp. 84–89.
- Cohen, D. An objective measure of seat comfort. *J. of Aviation, Space, & Env. Medicine*, 69, 4 (April 1998), 410–414.
- Cover, T.M. and Thomas, J.A.. *Elements of Information Theory*. Wiley Interscience, 1991.
- Cressie, N.A.C. *Statistics for Spatial Data*. Wiley, 1991.
- De Looze, M.P., Kuijt-Evers, L.M.F. and Van Dieen, J. Sitting comfort and discomfort and the relationships with objective measures. *Ergonomics*, 46, 10 (August 2003), 985–997.
- DiSalvo, C., Forlizzi, J., Zimmerman, J., Mutlu, B., and Hurst, A. The sensechair: The lounge chair as an intelligent assistive device for elders. In *Proc. of the AIGA Conf. on Designing for User Experiences*, 2005, 31.
- D’Mello, S.K., Craig, S.D., Gholson, B., Franklin, S., Picard, R., and Graesser, A.C. Integrating affect sensors in an intelligent tutoring system. In *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 Intl. Conf. on Intelligent User Interfaces*, 2005, pp. 7–13.
- Fitzgibbon, A.W., Pilu, M., and Fisher, R.B. Direct least-squares fitting of ellipses. *IEEE Trans. on Pattern Anal. & Machine Intel.*, 21, 5 (May 1999), 476–480.
- Fleuret, F. Fast Binary Feature Selection with Conditional Mutual Information. *J. on Machine Learning Research*, 5 (2004), 1531–1555.
- Friedman, J., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting, Stanford University, Dept. of Statistics Tech. Report, 1998.
- Furugori, S., Yoshizawa, N., Iname, C., and Miura, Y. Measurement of driver’s fatigue based on driver’s postural change. In *Proc. of SICE 2003 An. Conf.*, 1, 2003, pp. 264–269.
- Gonzalez-Banos, H.H., and Latombe, J. A randomized art-gallery algorithm for sensor placement. In *Proceedings of 17th ACM Symposium on Computational Geometry*, 2001, pp. 232–240.
- Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46 (2002), 389–422.
- Halender, M.G. and Zhang, L. Field studies of comfort and discomfort in sitting. *Ergonomics*, 40, 9 (1997), 895–915.
- Harada, T., Sato, T., and Mori, T. Estimation of bed-ridden human’s gross and slight movement based on pressure sensors distribution bed. In *Proc. of IEEE Intl. Conf. on Robotics & Automation*, 2002, pp. 3795–3800.
- Hossain, M., Morris, J.N., Brandeis, G.H., Ooi, W.L., and Lipsitz, L.A. A longitudinal study of risk factors associated with the formation of pressure ulcers in nursing homes. *J. of American Geriatric Society*, 42, 4 (1994).
- Kapoor, A., Mota, S., and Picard, R.W. Towards a learning companion that recognizes affect. In *Proc. of Emotional & Intelligent II: The Tangled Knot of Social Cognition, Fall Symposium*. AAAI, Nov. 2001.
- Krause, A. and Guestrin, C. A note on the budgeted maximization on submodular functions. Technical Report CMU-CALD-05-103, 2005.
- Krause, A. and Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *Proc. of Uncertainty in A.I.*, 2005.
- Landwehr, N., Hall, M., and Frank, E. Logistic model trees. In *Proc. of ECML/PKDD Conf.*, 2003, pp. 241–252.
- Monette, M., Weiss-Lambrou, R., and Dansereau, J. In search of a better understanding of wheelchair sitting comfort and discomfort. In *Proc. of the RESNA An. Conf.*, 1999, pp. 218–220.
- Mota, S. and Picard, R.W. Automated Posture Analysis for Detecting Learners Interest Level. In *Proc. of Computer Vision & Pattern Recognition Workshop*, 2003, pp. 49.
- Ozer, L.B. and Wolf, W. Real-time posture and activity recognition. In *Proc. of Workshop on Motion & Video Computing*, 2002, pp. 133–138.
- Panini, L. and Cucchiara, R. A machine learning approach for human posture detection in domotics applications. In *Proc. of Image Analysis & Processing*, 2003, pp. 103–108.
- Pellegrini, S. and Iocchi, L. Human Posture Tracking and Classification through Stereo Vision. In *Proc. of Intl. Conf. on Computer Vision Theory & Applications*, 2006.
- Prado, M., Reina-Tosina, J., and Roa, L. Distributed intelligent architecture for falling detection and physical activity analysis in the elderly. In *Proc. of EMBS/BMES Conf.*, 2002, pp. 1910–1911.
- Rosenfeld, A. and Pfaltz, J.L. Sequential operations in digital processing. *J. of ACM*, 13 (1966), 471–494.
- Sato, T., Nishida, Y., and Mizoguchi, H. Robotic Room: Symbiosis with human through behavior media. In *Robotics & Autonomous Systems*, Elsevier, 1996, pp. 185–194.
- Scherer, M.J. User desires for wheelchairs. *Rehab Management*, 9, 4 (June-July 1996), 121–123.
- Shaw, G. Wheelchair seat comfort for the institutionalized elderly. *Assistive Technology*, 3, 1 (1992), 11–23.
- Smith, D.M. Pressure ulcers in the nursing home. *Annals of Internal Medicine*, 123, 6 (Sept. 15, 1995), 433–438.
- Sviridenko, M. A note on maximizing a submodular set function subject to a knapsack constraint. *O. R. Letters*, 32, 1 (2004), 41–43.
- Tan, H.Z., Slivovsky, L.A., and Pentland, A. A sensing chair using pressure distribution sensors. *IEEE/ASME Trans. on Mechatronics*, 6, 3 (2001), 261–268.
- Tan, H. Z., Ifung L., and Pentland A. The chair as a novel haptic user interface. In *Proc. of Workshop on Perceptual User Interfaces*, Banff, Canada, 1997.
- Trivedi, M.M., Shinko, Y.C., Childers, E.M.C., and Krotosky, S.J. Occupant posture analysis with stereo and thermal infrared video: algorithms and experimental evaluation. *IEEE Trans. on Vehicular Technology*, 53, 6 (2004), 1698–1712.
- Witten, I.H. and Frank, E. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2005.
- Zhu, M., Martinez, A.M., and Tan, H.Z. Template-based recognition of static sitting postures. In *Workshop on Computer Vision & Pattern Recognition for Human-Computer Interaction*, 2003.