Predicting Daily Behavior via Wearable Sensors

Brian Clarkson and Alex Pentland

{clarkson, sandy}@media.mit.edu

We report on ongoing research into how to statistically represent the experiences of a wearable computer user for the purposes of day-to-day behavior prediction. We combine natural sensor modalities (camera, microphone, gyros) with techniques for automatic labeling from sparsely labeled data. We have also taken the next required step to build robust statistical models by beginning an extensive data collection experiment, the "I Sensed" series, a 100 day data set consisting of full surround video, audio, and orientation.

Keywords: contextual computing, peripheral sensing, Hidden Markov Models (HMM), computer vision, computer audition, wearable computing

I. Introduction

Is a person's day-to-day behavior predictable? We are concerned with this question because it is exactly the question that needs to be answered if we are to build agents (wearable or not) that anticipate. Agents that don't anticipate can react and reconfigure based on the present [1] and the past, but generally don't extrapolate into the future. This is a severe limitation because agents without predictive power cannot engage in preventive measures, "meet you half way", nor engage in behavior modification. This is not to say that a clever engineer couldn't herself notice a particular situation that is clearly indicative of some future state, and thus, manually program an agent to anticipate that future state when the situation occurs. However, definitely for a wearable agent and possibly others, the typical situations span the entire complex domain of real life where it is just unreasonable for anyone to manually design such anticipatory behavior into an agent. [2]

There are many ways to pose the question of predictability. In rough terms, prediction is being able to say with some level of certainty that if **A** happens then some time in the future **B** will happen. What we haven't specified yet is what domain is **A** and **B** coming from. There is a whole spectrum of possibilities for **A** and **B** that has to do with how detailed the agent's sensory input is. Can the agent understand what is being spoken and understand facial expression? Or, can it only know that there are speech-like sounds and something moving? The problem with these two ends of the spectrum of sensor detail is that sensor detail seems to be positively correlated with usefulness. It is our belief and purpose of this work that even at the lower end of sensor detail there are useful artificial intelligence systems that can be built, especially in such complex and rich domains as a wearable affords.

Theoretically, the question of how predictable a person's day-to-day behavior is moot if we have access to a complete description of the state of the world, right down to the electron spins in the user's fingernails. Then supposedly we can just apply the laws of physics and simulate into the future. The wearable that could do this would probably be quite uncomfortable to wear given current technology. Another approach is to start with the coarsest description of the state of the world, see what can be deduced from it and then move to a slightly finer description. You stop when the size of the wearable or its level of privacy invasion outweighs the benefits it delivers.

In this work we will take a straightforward approach to answering this question of predictability. First, we will address the problem of building coarse descriptions of the world from wearable sensors, such as cameras, microphones, and gyros. Then we will report on an extensive data collection experiment that allows us to build predictive models of a person's day-to-day behavior.

II. Automatic Situation Segmentation

The goal at hand is to reliably learn and classify the user's situation from as few labeled examples as possible [3]. We discuss this overall task in terms of the subtask of location recognition since it is well-defined and performance is easily evaluated [4, 5]. However, having a small self-contained sensored device that can gradually and automatically learn the various states or conditions of its environment is of general importance to many tasks. From cellular phones and wearable computers to robots and smart rooms, many situations/applications could benefit from this kind of a smart sensor. [6]

Before attempting to minimize the number of labeled examples required to train a location classifier, we first evaluate the performance we can achieve on the data set under typical training conditions. The pipeline for regular location classification starts with a feature extraction step. For each location, HMMs were estimated from a training set and then probability measurements were derived from the log likelihoods on a test set.

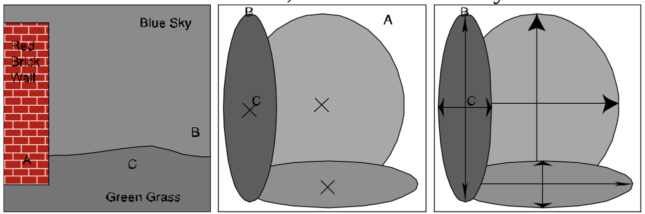
A. Video Feature Set

For the video images, we calculate spatial moments in each of the color channels, Y, Cb and, Cr, as follows:

$$M_{c,m,n}[t] = \frac{\sum_{i=0}^{H} \sum_{j=0}^{W} i^m j^n P_{c,i,j}[t]}{H^m W^n \sum_{i=0}^{H} \sum_{j=0}^{W} P_{c,i,j}[t]}$$

$$(c,m,n) \in \{Y,U,V\} \times \{0,1\} \times \{0,1\}$$

$P_{c,i,j}[t]$ is the value of the color channel, *c*, for the pixel, *(i,j)* and *H* and *W* are the image extents. This yields a 12 dimensional feature vector, 3 color channels by 4 moments.



$m=0$ $n=0$	$m=1$ & $m=0$ $n=0$ & $n=1$	$m=1$ $n=1$

The 4 types of image moments (as determined by the exponent of the pixel location) measure 3 aspects of the spatial pixel distribution: mass, geometric center, and geometric spread. For example the figure shows an abstract image with its dominating pixel distributions in each color channel. In each panel a different property of these distributions are measured and shown. [7]

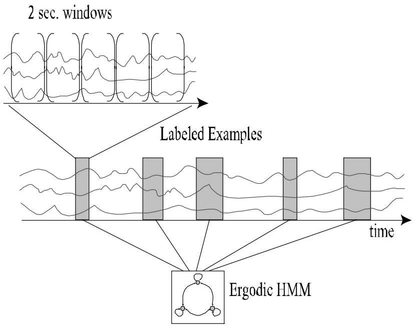
B. Audio Feature Set

For the audio, we simply calculate a spectrogram using a 1024-pt FFT at 15Hz. The spectrogram was passed through a bank of Mel-scale filters to yield 11 coefficients per unit time. The resulting time sequence of spectral coefficients was then low-pass filtered with a single-pole IIR filter with a time constant of 0.4 seconds:

$$y[t] = 0.999 y[t-1] + x[t]$$

and subsequently sampled at 5Hz. A similarly low-passed filtered estimate of energy is also calculated for a grand total of 12 auditory features. [8] [9]

C. Training Models



These features, $\{x_t\}$, were modeled by ergodic Hidden Markov Models (HMM). To train the HMMs, each example of a location was divided into 2 sec. windows (or equivalently, $N=10$ feature vectors at 5Hz) of features. These windows of features were gathered into one set and used to train a class HMM (with Expectation-Maximization). By design, the class HMMs, when trained in this way, end up modeling the "dynamic texture" of a location at a 2 second time-scale. [10]

D. Model Evaluation

At this point we can just construct a classification grammar from all of the location HMMs and use Viterbi to solve for the most likely segmention. We have done this and it works well, but we would like to just evaluate each model independently for its absolute 2-class probability (true or false). This methodology makes the inclusion of these models in larger (inhomogeneous and incremental) learning frameworks much easier.

So, to determine for each time, *t*, the probability of a given location, the Forward-Backward algorithm was used to yield,

$$\log P(x_{t-N},...,x_t \mid y=1)$$

When using the trained HMMs to evaluate the probability of a class given a window of features, $P(y \mid x_{t-N},...,x_t)$, it is necessary to estimate:

$$P(x_{t-N},...,x_t) = P(x_{t-N},...,x_t \mid y=1)P(y=1) + P(x_{t-N},...,x_t \mid y=0)P(y=0)$$

(Notice we are not assuming that at any given point in time only one class can be active.) The first term is given by the Forward-Backward algorithm, but the second term is not available to us. Training a second HMM (i.e. a garbage model) on the training data that is not labeled as being part of the class has been tried. However, this training set is in most cases utterly incomplete and hence the garabage HMM does not model everything outside of the given class. So when data outside of the training set is encountered, the garbage model's likelihood often drops, artificially and incorrectly increasing the class probability.

So we are still left with the problem of recovering a class probability from the HMM log likelihood. Approaching the problem from a totally different perspective. When thresholding probabilities, the correct MAP threshold for 2-class problems is $p_{threshold} = 0.5$. Fortunately, there is a prinicipled manner for determining,

$$l_{threshold} = \phi^{-1}(p_{threshold}) = \phi^{-1}(0.5)$$

and that is by the Receiver-Operator Characteristic (ROC). The log likelihood threshold that achieves the Equal Error Rate (EER) point on the ROC curve is the value that should be mapped to a probability of 0.5. The mapping for the 2 remaining intervals, [0 0.5) and (0.5, 1], by histogramming the log likelihoods in each set and calculating the cumulative distribution function (cdf). We took these 2 cdf's and concatenated them to produce one continuous mapping function, ϕ, that assigned all log likelihoods to [0, 1] with $\phi(l_{threshold}) = 0.5$. The result is a pseudo-probability, $P(y \mid x_{N-t},...,x_t)$, that is appropriate for inter-model comparison.

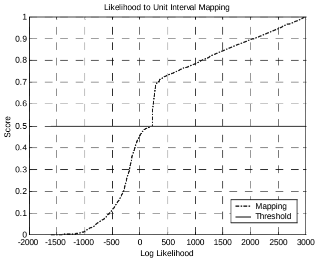


Figure 1: The histogram based mapping from raw log likelihood to a probability score.

	Correct Acceptance			Correct Rejection		
Locations	**A+V**	**A**	**V**	**A+V**	**A**	**V**
BorgLab	95.9	19.1	97.1	92.1	56.2	84.9
BTLab	93.3	63.8	88.3	97.3	48.0	98.8
Courtyard	83.1	38.2	93.0	92.2	64.9	76.6
Elevator	63.6	52.1	62.8	99.8	58.0	98.4
Lower Atrium	95.7	88.7	87.3	60.9	26.8	56.3
Upper Atrium	95.0	56.3	96.0	60.7	52.3	61.4
Office	89.9	42.6	71.1	96.0	87.3	93.5

Table 1: Baseline Recognition Results

E. One Shot Learning or Automatic Labeling

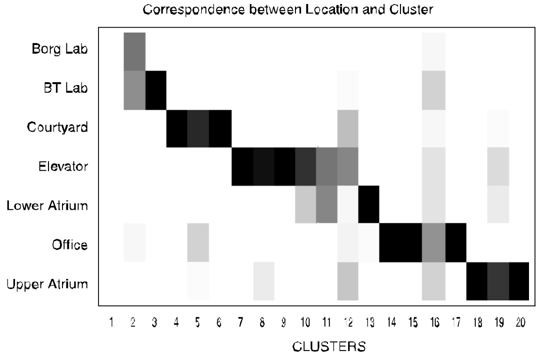
Typically, we cannot expect the user to spend hours to collect and label the amount of data that the above procedure requires to build robust models. Instead we need to minimize the amount of labeling required. Maximizing our ability to incorporate prior information can do exactly that. Our methodology was to use a small amount of labeled data to seed models or clusters and then use prior information as (soft) constraints that allowed us to implicitly label a large amount of unlabeled data. This kind of framework naturally supports incremental (and hence adaptive) learning.

F. Generalized Clustering

Clustering (such as K-Means) finds the natural division of the data that is supported by your centroid models. In the typical application of K-Means clustering, the centroid model is a Gaussian, however, it could also be an HMM (as in Segmental K-Means and its variations). Generalized K-Means for any centroid model, M, is implemented as follows:

```
Randomly initialize $K$ models, $\{M_1,...,M_K\}$.
For each data vector, $x$, in the data set:
        calculate: $M_{\max} = \arg\max_{M \in \{M_i\}} \{P(x \mid M)\}$
        assign $x$ to $M_{\max}$
Update each model in $\{M_1,...,M_K\}$, & repeat.
```

Now suppose the centroid model, M, was also being used for a recognition task, and K >> number of classes. If a label of the training set lies entirely within a given cluster then it follows that we can extend that label to the rest of the data in the cluster without detrimentally affecting the performance, but perhaps increasing the robustness of the recognition. The more valid the centroid model, M, is for the recognition task, the closer K can be to the actual number of classes and hence the fewer labels we need to label all of the data. [11] [12]



G. Generalized Change Detection

W can also obtain a useful segmentation of unlabeled data by a method we call generalized change detection. In general, a feature, x, is measured at each time, t, and then a measure of change, such as $\Delta x = x[t] - x[t-1]$ in the simplest case, is used to detect areas of likely scene changes. However, with high frequency (i.e. noisy) features a direct measure of change is not useful due to a chronic problem of overfiring. Therefore, a more general change detection measure is one that actually tries to measure the probability of a scene change at every given time given some model of feature dynamics or noise [13].

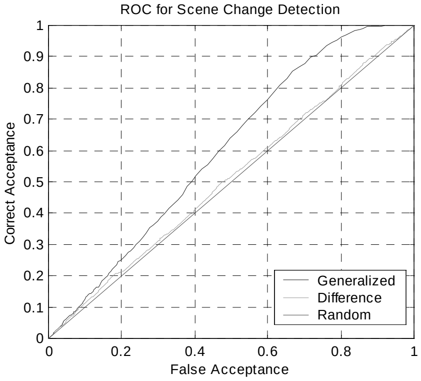
Here we can take advantage of the K models, $\{M_1,...,M_K\}$, that were obtained from the Generalized Clustering algorithm. If the cluster model is an HMM then each of these cluster models, by design, represents a bit of feature "behavior". From the probabilities of these K models at a time, t, we can construct a K-dimension activation vector:

$$A[t] \triangleq \begin{bmatrix} P(M_1 \mid x_{N-t},...,x_t) \\ \vdots \\ P(M_K \mid x_{N-t},...,x_t) \end{bmatrix}$$

Now we can go back to a natural measure of change to detect scene changes:

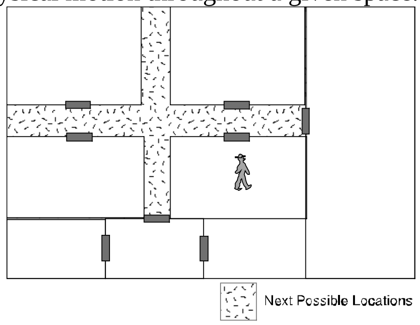
$$\Delta A = \left\| A[t] - A[t-1] \right\|$$

This effectively measures the change in the distribution, or composition, of cluster models that are representing the features around time, t.

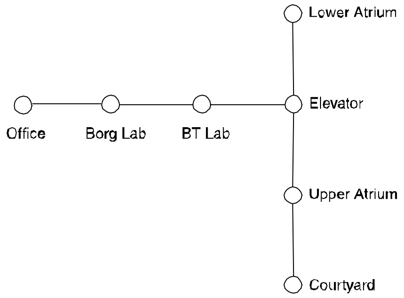


H. Spatial Constraints

A commonly-used and powerful source of prior information in location recognition is simply a map of the area. A geographic map encodes the Markov constraints that govern physical motion throughout a given space.



A table of conditional probabilities, $P(y_t = i \mid y_{t-1} = j)$, where $i, j \in \{locations\}$, (even an approximate one) can greatly constrain and thus boost the performance of the regular location classification. More importantly it could allow us to solve for the correspondence between locations and unlabeled clusters in the data and thus greatly reduce the amount of labeled data that training requires. Here is the geographic constraint network for the location recognition subtask, its use boosts the recognition rates for all locations to 100%:



III. Data Collection

We have described our relevant experiments and methods for chunking up wearable sensor streams into salient events and situations. The next phase in our quest to answer the question of human predictability is to accumulate a series of these events and situations experienced by one person over an extended period of time.

A. The "I Sensed" Series: 100 Days of Experiences

The first requirement of learning predictive models from data is to have enough repeated trials of the experiment from which to estimate robust statistics. Ideally experiential data recorded from an individual over a number of years would be ideal. However, other forces such as the computational and storage requirements needed for huge data sets force us to settle for something smaller, but not too small. We chose 100 days (14.3 weeks) because, while it is a novel period for a data set of this sort, its size is still computationally tractable (approx. 500 gigabytes).

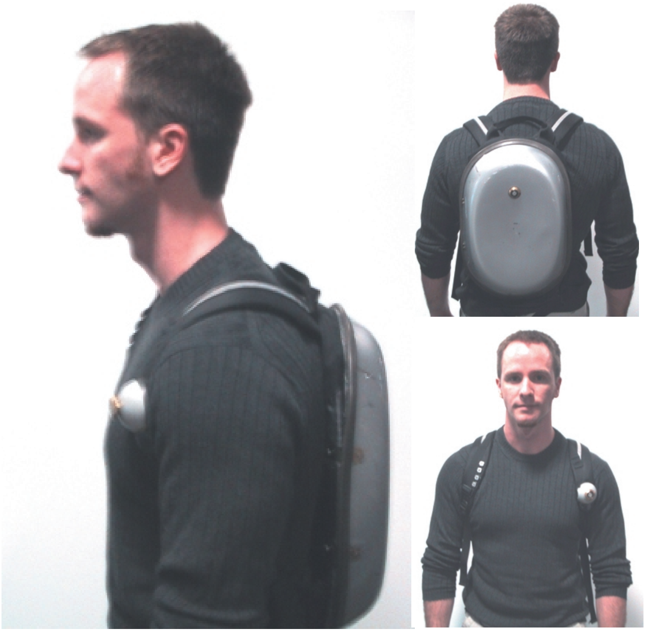


Figure 2: The Data Collection wearable when worn.

Due to the lack of volunteers for this experiment, it has been left up to the author (Brian Clarkson) to perform this data collection on himself. As of the writing of this paper, I have been collecting data continuously for the last 30 days. The "I Sensed" series, the name of the data set (inspired by conceptual artist Kawara On), is scheduled to be completed in mid-July of 2001. Refer to the last page (Figure 5) of this paper for actual excerpts from this data set on 4 different scenes: eating lunch, walking up stairs, in a conversation, and rollerblading.

The parameters of the experiment are as follows:

- Data collection commences each day from approx. 10am and continues until approx. 10pm. This varies based on the sleeping habits of the experimental subject.
- The times that the data collection system is not active or worn by the subject is logged and recorded. Such times are typically when: batteries fail, sleeping, showering, and working out.
- In addition to the visual, aural, and orientational sensor data collected by the wearable, the subject is also required to keep a rough journal of his high-level activities to within the closest half hour. Examples of high-level activity are: "Working in the office", "Eating lunch", "Going to meet Michael", etc. while being specific about who, where, and why.
- Every 2 days the wearable is "emptied" of its data, by uploading to a secure server.
- Persons who normally interact with the subject on a day-to-day basis and have a possibility of having a potentially private conversation recorded are asked to sign a consent form and an agreement is made by the experimenters to not disclose the data in way without further consent.

B. The Data Collection Wearable

The sensors chosen for this data set are meant to mimic the human senses. They include visual (2 camera, front and back), auditory (1 microphone), and gyros (for 3 degrees of orientation: yaw, pitch and roll). These match up with the eyes, ears, and inner ear, while taste and smell are not covered because the technology is not available yet. Other possibilities for sensors that have no good reason for being excluded are temperature, humidity, accelerometers, and bio-sensors (e.g. heart-rate, galvanic skin response, glucose levels). The properties of the 3 sensor modalities are as follows: (see Figure 3)

Audio: 16kHz, 16bits/sample (normal speech is generally only understandable for persons in direct conversation with the subject.)

Front Facing Video: 320x240 pixels, 10Hz frame rate (faces are generally only recognizable under bright lighting conditions and from less than 10ft away.)

Back Facing Video: 320x240 pixels, 10Hz frame rate (faces are generally only recognizable under bright lighting conditions and from less than 10ft away.)

Orientation: Yaw, roll, and pitch are sampled at 60Hz. A zeroing switch is installed beneath the left strap which is meant to trigger whenever the subject puts on the wearable. Drift is only reasonable for periods of less than a few hours.

The wearable is based on a backpack design for comfort and wardrobe flexibility. The visual component of the wearable consists of 2 USB cameras (front- and rear-facing) modified to be optically compatible with 200° field-of-view lenses (adapted from door viewers). This means that we are recording light from every direction in a full sphere around the user (but not with even sampling of course). The front-facing camera is sewn to the front strap of the wearable and the rear-facing camera is contained inside the main shell-like compartment. The microphone is attached directly below the front-facing camera on the strap. The orientation sensor (InterTrax2 from Intersensed Inc. with its magnetic field zeroing feature disabled) is housed inside the main compartment. Also in the main compartment are a PIII 500MHz cell computer (CellComputing Inc.) with a 10GB HDD (enough storage for 2 days) and 4 Sony Infolithiums NP-F960 (operating time: ~10 hrs.). The polystyrene shell (see Figure 2) was designed and vacuum-formed to fit the components as snuggly as possible while being aesthetically pleasing, presenting no sharp corners for snagging, and allowing the person reasonable comfort while sitting down.

Since this wearable is only meant for data collection, its input and display requirements are minimal. For basic on/off, pause, record functionality there are click buttons attached to the right-hand strap (easily accessible by the left-hand by reaching across the chest). These buttons are chorded for protection against accidental triggering. All triggering of the buttons (intentional or otherwise) is recorded along with the sensor data. Other than the administrative functions, the buttons also provide a way for the subject to mark salient points in the sensor data. The only display provided by the wearable is 2 LEDs, one for power and the other for recording.

C. The Data Journal

Recently, we have completed the system that allows us fully transcribe the “I Sensed” series and access it arbitrarily in a multiresolution manner. This ability is essential for clustering and scene change detection techniques talked about in the first section of this paper. All data (images, frames of audio, button presses, orientation vectors, etc.) are combined and time synchronized in the Data Journal system to millisecond accuracy (see Figure 4).

IV. CONCLUSIONS

We have reported on and shown the feasibility of one-shot learning techniques for automatically segmenting wearable sensor data into scenes. We combine various approaches including knowledge engineering techniques such as encoding geographic constraints from a map, clustering techniques such as clustering with HMMs, and generalized change detection for improved scene change estimation. These techniques provide the backdrop for the extended data set we are currently collecting, the “I Sensed” series. This 100 day data set when segmented for scenes and events will allow us to build predictive models for the subject’s day-to-day behavior.

Note to reviewers: As we wait for more of the “I Sensed” series to be completed we will be able to generate results concerning predictive models of this new data set. These results will be added to the camera-ready version. Also the location recognition results will be generalized to a wider variety of situations.

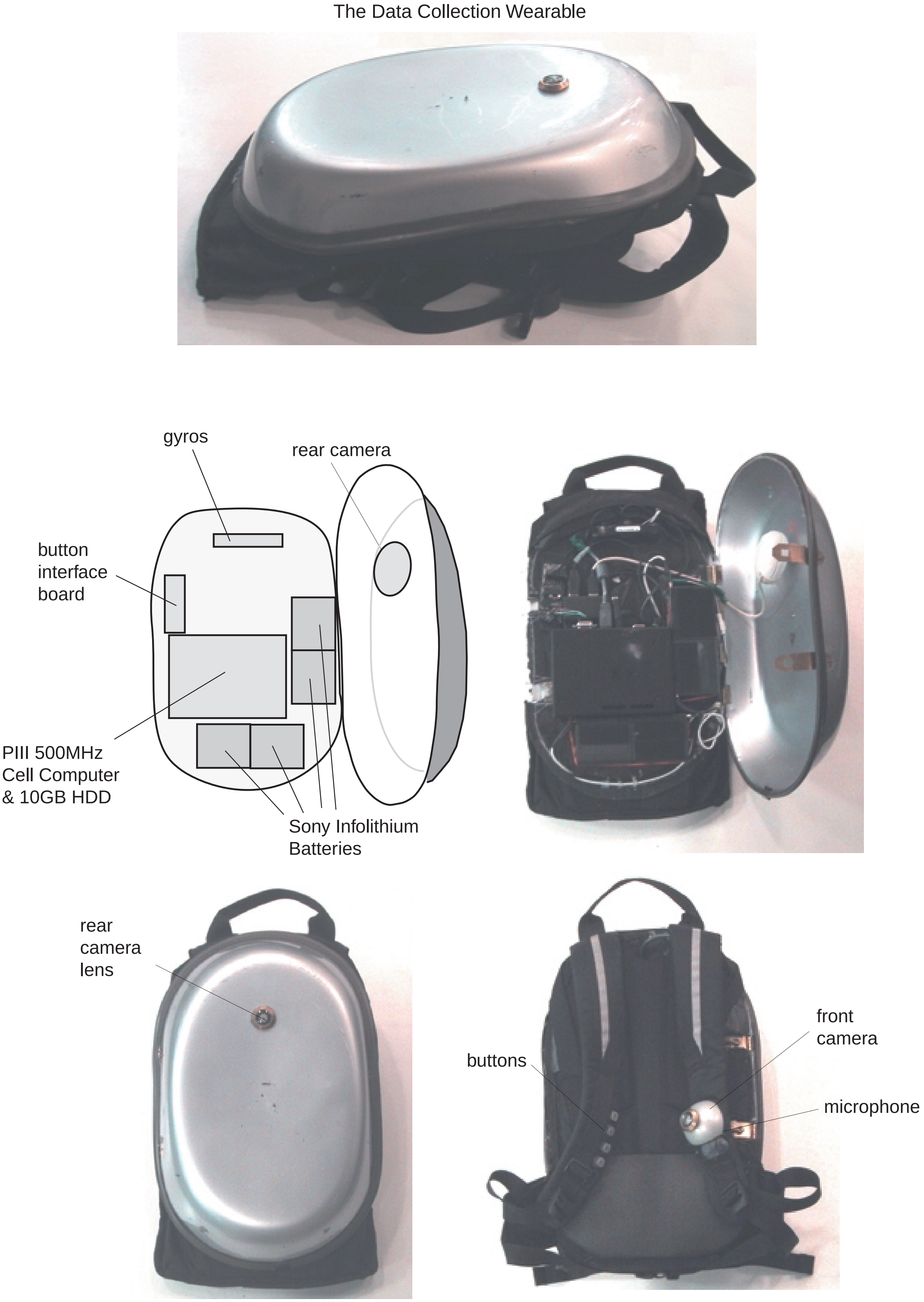

The Data Collection Wearable

Figure 3: The Data Collection Wearable Schematic

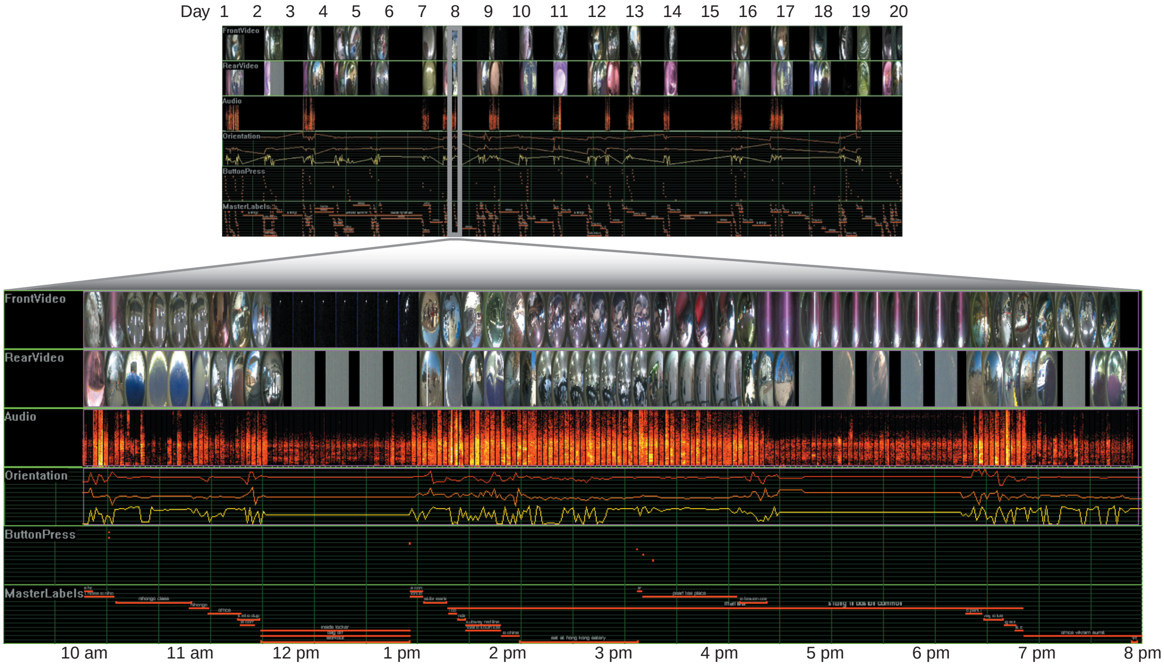


Figure 4: The Data Journal System: provides a multiresolution representation of the time-synchronized sensor data.

V. Bibliography

1. Aoki, H., B. Schiele, and A. Pentland, *Realtime Personal Positioning System for Wearable Computers.* International Symposium on Wearable Computers '99, 1999.
2. Blum, A. and T. Mitchell, *Combining Labeled and Unlabeled Data with Co-Training.* Proceedings of the 1998 Conference on Computational Learning Theory, 1998.
3. Clarkson, B. and A. Pentland, *Unsupervised Clustering of Ambulatory Audio and Video*, in *ICASSP'99*. 1999: http://www.media.mit.edu/~clarkson/icassp99/icassp99.html.
4. Clarkson, B., K. Mase, and A. Pentland, ***Recognizing User's Context from Wearable Sensor's: Baseline System***. Vismod Technical Report #519, 2000.
5. Feiten, B. and S. Gunzel, *Automatic Indexing of a Sound Database Using Self-organizing Neural Nets.* Computer Music Journal, 1994. **18**(Fall): p. 53-65.
6. Foote, J., *A Similarity Measure for Automatic Audio Classification*. 1997, Institute of Systems Science.
7. Lamming, M. and M. Flynn, *"Forget-me-not" Intimate Computing in Support of Human Memory*. 1994.
8. Lieberman, H. and D. Maulsby, *Instructible Agents: Software that just keeps getting better*. IBM Systems Journal, 1996. **35**(3&4): p. 539-556.
9. Lin, T. and H.-J. Zhang, *Automatic Video Scene Extraction by Shot Grouping.* International Conference on Pattern Recognition, 2000. **4**.
10. Liu, Wang, and Chen, *Audio Feature Extraction and Analysis for Multimedia Content Classification.* Journal of VLSI Signal Processing Systems, 1998(June).
11. Mannila, H., H. Toivonen, and A.I. Verkamo, *Discovery of freqruent episodes in event sequences.* Series of Publications C, 1997(Report C-1997-15).
12. Mindru, F., T. Moons, and L.v. Gool, *Recognizing color patterns irrespective of viewpoint and illumination.* CVPR99, 1999: p. 368-373.
13. Nakamura, Y., J.y. Ohde, and Y. Ohta, *Structuring Personal Activity Records based on Attention -- Analyzing Videos from Head-mounted Camera.* International Conference on Pattern Recognition, 2000. **4**: p. 222-225.
14. Nakamura, Y., J.y. Ohde, and Y. Ohta, *Structuring Personal Activity Records based on Attention -- Analyzing Videos from Head-mounted Camera.* International Conference on Pattern Recognition, 2000. **4**: p. 222-225.
15. Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 1989. **77**(February): p. 257-284.
16. Starner, T., B. Schiele, and A. Pentland. *Visual Contextual Awareness in Wearable Computing*. in *Second International Symposium on Wearable Computers*. 1998.
17. Zhong, D., H.J. Zhang, and S.-F. Chang, *Clustering Methods for video browsing and annotation.* SPIE, 1996(Storage and Retrieval for Still Image and Video Databases IV).

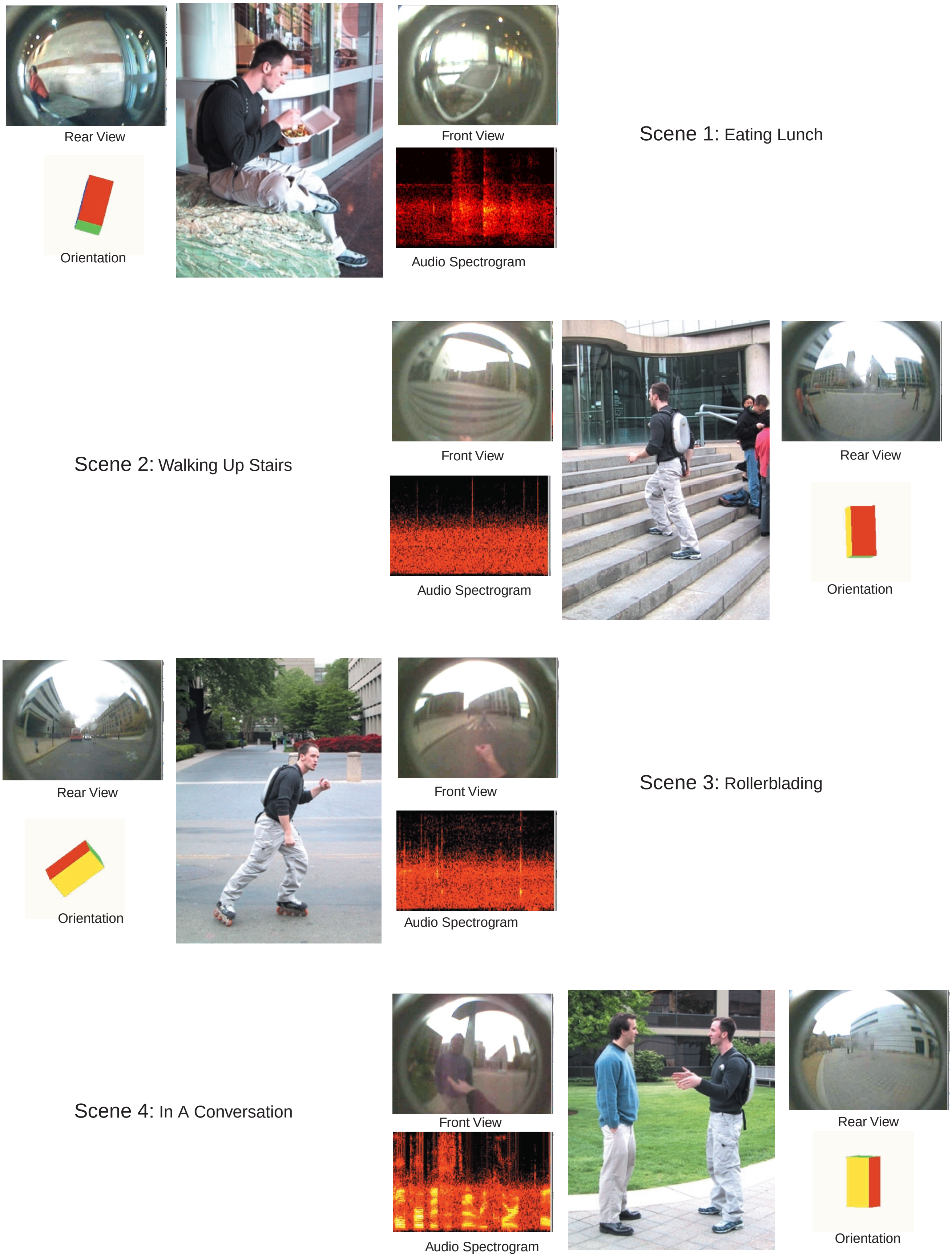


Figure 5: Some excerpts from the "I Sensed" series